Tampere University
of Technology

WIAMIS 2001

Workshop on Image Analysis for Multimedia Services
16 - 17 May 2001
Tampere, Finland

Edited by
Moncef Gabbouj
Tampere University of Technology

WIAMIS 2001 Proceedings
Printed in Finland
Printed in TTKK-Paino, Tampere, Finland

ISBN: 952-15-0625-3

The Chairman's Message

Following the tradition set at Louvain-la-Neuve in 1997 and Berlin in 1999, Tampere is honoured to host WIAMIS 2001. The Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) is sponsored by the EU COST 211 quat Action, Tampere International Center for Signal Processing and Tampere University of Technology. WIAMIS is a major window to the outside world for the closed collaborative Action COST 211. Cross fertilization of ideas is our prime goal in this workshop, in addition to attracting new members working on related areas to COST 211.

The current phase, COST 211 quat, focuses on two important aspects of multimedia interactive services, namely, moving sequence segmentation and content-based indexing, browsing and retrieval. Some 30 papers from 13 countries will be presented at WIAMIS 2001 covering both of these topics.

WIAMIS 2001 features two invited keynote speakers. Dr. Thomas Sikora, from Heinrich-Hertz Institute in Berlin and Chairman of the Video Group in MPEG-4, will talk about the next generation user interface technology. Professor Philippe Salembier, from Universitat Politecnica de Catalunya and Chairman of the MPEG-7 Mulmedia Description Group, will give an overview of the MPEG-7 standard and of future challenges for visual information analysis.

A special session is organized during Day 1 of the workshop where two successful proposals, in reply to the Call for Comparison with the COST AM which was launched by COST 211 quat, will be presented. The remaining papers are organized in oral and poster sessions.

I would like to thank all authors for their great efforts in making this high quality technical program. Sincere thanks are due to the members of the Technical Program Committee, in particular Geoff Morrison of BT, Ferran Marques and Philippe Salembier of UPC, Ebroul Izquierdo of QMUL and Thomas Sikora of HHI. Dr. Eric Badique, from the European Commission, is acknowledged for his support and encouragement, especially with the Call for Comparison with the COST AM. Assistance with the local arrangementsby Pasi Reijonen and Elina Orava from Tampere University of Technology is greatly appreciated.

Last but not least, I wish to thank our two sponsors: the COST Telecommunication Framework of the European Commission, in particular COST 211 quat Action and Tampere International Center for Signal Processing for sponsoring the workshop.

I wish all participants a fruitful and enjoyable workshop and a wonderful stay in Tampere.

Moncef Gabbouj
Chairman
WIAMIS 2001

Steering Committee

Chairman:
Moncef Gabbouj
Tampere University of Technology
Signal Processing Laboratory
P.O. Box 553
FI-33101 Tampere
Finland
Moncef.Gabbouj@tut.fi

Technical Program Committee
Jaakko Astola, *Tampere University of Technology*, Finland
Andrew Bangham, *University of East Anglia*, United Kingdom
Dominique Barba, *Université de Nantes*, France
Michel Barlaud, *Université de Nice – Sophie Antipolis*, France
Ebroul Izquierdo, *Queen Mary, University of London*, United Kingdom
Benoît Macq, *Universite Catholique de Louvain*, Belgium
Ferran Marquès, *Universitat Politecnica de Catalunya*, Spain
Geoff Morrison, *British Telecom*, United Kingdom
Fernando Pereira, *Instituto de Telecomunicações*, Portugal
Thomas Sikora, *Heinrich-Hertz Institute*, Germany
John R. Smith, *IBM T.J. Watson Research*, USA
Michael Strintzis, *Aristotle University of Thessaloniki*, Greece
Jukka Yrjänäinen, *Nokia Research Center*, Finland

Workshop Management
SuviSoft Oy Ltd.
Herminankatu 3, 33720 Tampere, Finland
suvisoft@suvisoft.fi

Local Arrangements
Pasi Reijonen, *Tampere University of Technology*, Finland
Elina Orava, *Tampere University of Technology*, Finland

TABLE OF CONTENT

Wednesday 16 May 2001

12:10 - 13:30 LUNCH

13:30 - 14:30 Multimedia Analysis I
Chair: Ebroul Izquierdo, Queen Mary University of London

14:30 - 15:00 COFFEE BREAK

15:00 - 16:20 Multimedia Analysis II
Chair: Geoff Morrison, BTexat Technologies

Thursday 17 May 2001

09:00 - 09:40 Invited Key Note Address 2: MPEG-7: Current Status and Future Challenges
Chair: Moncef Gabbouj, Tampere University of Technology

09:40 - 10:20 Multimedia Indexing and Retrieval I
Chair: Noel Murphy, *Dublin City University, Ireland*

10:20 - 10:50 COFFEE BREAK

10:50 - 11:10 Multimedia Indexing and Retrieval I (Cont'd)
Chair: Noel Murphy, *Dublin City University, Ireland*

11:10 - 14:00 Poster Session **(12.00 – 13.00 LUNCH)**
Chair: Faouzi Alaya Cheikh, Tampere University of Technology

14:00 - 14:40 Multimedia Indexing and Retrieval II
Chair: Philippe Salembier, *Universitat Politecnica de Catalunya, Spain*

14:40 - 15:00 COFFEE BREAK

15:00 - 16:20 Content Understanding and Analysis
Chair: Moncef Gabbouj, Tampere University of Technology

Sensing People - Next Generation User Interface Technology

Thomas Sikora

Heinrich-Hertz-Institute (HHI) for Communication Technology, Germany

sikora@hhi.de

Smart environments, wearable computers, perceptual user interfaces and ubiquitous computing are widely thought to be the coming "fourth generation" computing, information and communication technology. Before this new generation of computing can be widely deployed, it has to be equipped with "smart" sensing technology and user interfaces, that allow the computers to be used without detailed instructions and to respond to their environment automatically. At HHI, of prime importance are research topics related to the machine vision problem of detecting, tracking and identifying people. More specifically, our work is dedicated towards detecting humans, interpreting human behaviour, to understand where people are looking or pointing at, what task they are doing, and whether they are communicating. The purpose of the presentation at WIAMIS'01 is to introduce the "Sensing People" vision and to provide examples of applications scenarios.

VIDEO SEGMENTATION USING FAST MARCHING AND REGION GROWING ALGORITHMS

E. Sifakis, I. Grinias, and G. Tziritas
Dept. of Computer Science, University of Crete,
P.O.Box 2208, Heraklion, GREECE
Tel: +30 81 393517; fax: +30 81 393501
e-mail: {sifakis, grinias, tziritas}@csd.uoc.gr

ABSTRACT

The algorithm presented in this paper was proposed for comparisons using the COST 211 data set. It is comprised of three main stages: (1) classification of the image sequence, and parametric motion estimation in case of a moving camera, (2) change detection having as reference a fixed frame, an appropriately selected frame or a displaced frame, and (3) object localisation using local colour features. The image sequence classification is based on statistical tests on the frame difference. The change detection module uses the two-label fast marching algorithm. Finally, the object localisation uses a region growing algorithm based on the colour similarity.

1 INTRODUCTION

Video segmentation is a key step in image sequence analysis and its results are extensively used for determining motion features of scene objects, as well as for coding purposes to reduce storage requirements. The development and wide-spread use of the international coding standard MPEG-4 [11], which relies on the concept of image/video objects as transmission elements, has raised the importance of these methods. Moving objects could also be used for content description in MPEG-7 applications.

Various approaches have been proposed for video or spatio-temporal segmentation. An overview of segmentation tools, as well as of region-based representations of image and video, are presented in [6]. The video object extraction could be based on change detection and moving object localisation, or on motion field segmentation, particularly when the camera is moving. Our approach is based exclusively on change detection. The costly and potentially inaccurate motion estimation process is not needed. We present here some relevant work from the related literature for better situating our contribution.

In the framework of COST 211 an Analysis Model (AM) is proposed for image and video analysis and segmentation [2]. The essential feature of the AM is its ability to fuse information from different sources: colour segmentation, motion segmentation, and change detection. Kim *et al.* [5] proposed a method using global motion estimation, change detection, temporal and spatial segmentation.

Our algorithm, after the global motion estimation phase, is mainly based on change detection. The change detection problem is formulated as two-label classification. In [8] we have introduced a new methodology for pixel labelling called *Bayesian Level Sets*, extending the *level set* method [7] to pixel classification problems. We have also introduced the *Multi-label Fast Marching* algorithm and applied it at first to the change detection problem [10]. A more recent and detailed presentation is given in [9]. The algorithm presented in this paper differs from previous work in the final stage where the boundary based object localisation is replaced by a region based object labelling.

In Section 2 the method for selecting the appropriate frame difference for detecting the moving object is presented. In Section 3 we present the multi-label fast marching algorithm, which uses the frame difference and an initial labelling for segmenting the image into unchanged and changed regions with respect to the camera, *i.e.* changes independent of the camera motion. The last step of the entire algorithm is presented in Section 4 where a region growing technique extends an initial segmentation map. Section 5 concludes the paper, commenting on the obtained results.

2 FRAME DIFFERENCE

In our approach the main step in video object segmentation is change detection. Therefore for each frame we must first determine another frame which will be retained as reference frame and used for the comparison. Three different main situations may occur: (a) a constant reference frame, as in surveillance applications, (b) another frame appropriately selected, in the case of a still camera, and (c) a computed displaced frame, in the case of a moving camera.

The image sequence must be classified according to the above categories. We use a hierarchical categorization based on statistics of frame differences. At first the hypothesis (a) is tested against the other two. We can consider there to exist a unique background reference

image if, for a number of frames, the observed frame differences are negligible. A test on the empirical probability distribution is then used.

When the reference is not constant we have to determine the more appropriate reference in order to identify independently moving objects. In order to determine the reference frame, it must be decided if the camera is moving or not. The test is again based on the empirical probability distribution of the frame differences.

Before considering the two possible cases we will present the statistical model used for the frame difference, because the determination of the appropriate reference frame is based on this model. Let $D = \{d(x,y), (x,y) \in S\}$ denote the gray level difference image. The change detection problem consists of determining a "binary" label $\Theta(x,y)$ for each pixel on the image grid. We associate the random field $\Theta(x,y)$ with two possible events, $\Theta(x,y) = \text{static}$ (*unchanged pixel*), and $\Theta(x,y) = \text{mobile}$ (*changed pixel*). Let $p_{D|\text{static}}(d|\text{static})$ (resp. $p_{D|\text{mobile}}(d|\text{mobile})$) be the probability density function of the observed inter-frame difference under the H_0 (resp. H_1) hypothesis. These probability density functions are assumed to be zero-mean Laplacian for both hypotheses ($l = 0, 1$)

$$p(d(x,y)|\Theta(x,y) = l) = \frac{\lambda_l}{2} \quad e^{-\lambda_l |d(x,y)|}. \qquad (1)$$

Let P_0 (resp. P_1) be the *a priori* probability of hypothesis H_0 (resp. H_1). Thus the probability density function is given by

$$p_D(d) = P_0 \; p_{D|0}\,(d|\text{static}) + P_1 \; p_{D|1}(d|\text{mobile}). \qquad (2)$$

In this mixture distribution $\{P_l, \lambda_l; l \in \{0,1\}\}$ are unknown parameters. The principle of Maximum Likelihood is used to obtain an estimate of these parameters [3].

In the case of a still camera, the current frame must be compared to another frame sufficiently distinct, *i.e.*, is a frame where the moving object is displaced to be clearly detectable. For that the mixture of Laplacian distributions (2) is first identified. The degree of discrimination of the two distributions is indicated by the ratio of the two corresponding standard deviations, or, equivalently, by the ratio of the two estimated parameters λ_0 and λ_1. So we search for the closest frame, which is sufficiently discriminated from the current one. The threshold (T_λ) on the ratio of standard deviations is supplied by the user, and thus is determined the frame difference.

In the case of a moving camera the frame difference is determined by the displaced frame difference of successive frames. The camera movement must be computed for obtaining the displaced frame difference. We use a three-parameter model for describing the camera motion, composed of two translation parameters and a zoom parameter. The estimation of the three parameters is based on a frame matching technique with a robust criterion of least median of absolute displaced differences. For computational complexity reasons the median is determined using the histogram of the absolute displaced frame differences.

3 CHANGE DETECTION USING FAST MARCHING ALGORITHM

3.1 Initial labelling

An initial map of labelled sites is obtained using statistical tests. The first test detects changed sites with high confidence. The false alarm probability is set to a small value, say P_F. For the entire COST data set $P_F = 10^{-7}$. Subsequently a series of tests is used for finding unchanged sites with high confidence, *i.e.*, with a small probability of non-detection. For these tests a series of six windows of dimension $(2w+1)^2$, $w = 2, \ldots, 7$, is considered and the corresponding thresholds are preset as a function of λ_1. Let us denote by B_w the set of pixels labelled as unchanged when testing window indexed by w. We set them as follows

$$B_w = \{(x,y) : \sum_{k=-w}^{w} \sum_{l=-w}^{w} |d(x+k, y+l)| < \frac{\gamma_w}{\lambda_1}\},$$

for $w = 2, \ldots, 7$. The probability of non-detection depends on the threshold γ_w, while λ_1 is inversely proportional to the dispersion of $d(x,y)$ under the "changed" hypothesis. As the evaluation of this probability is not straightforward, the numerical value of γ_w is empirically fixed. Finally the union of the above sets $\cup_{w=2}^{7} B_w$ determines the initial set of "unchanged" pixels.

3.2 Label propagation

A multi-label fast marching level set algorithm is then applied to all sets of points initially labelled. This algorithm is an extension of the well-known fast marching algorithm [7]. The contour of each region is propagated according to a motion field, which depends on the label and on the absolute inter-frame difference. The label-dependent propagation speed is set according to the *a posteriori* probability principle. As the same principle will be used later for other level set propagations and for their respective velocities, we shall present here the fundamental aspects of the definition of the propagation speed. The candidate label is ideally propagated with a speed in the interval $[0,1]$, equal in magnitude to the *a posteriori* probability of the candidate label at the considered point. Let us define at a site (x,y), for a candidate label l and for a data vector d the propagation speed as

$$v_l(x,y) = \Pr\{l(x,y)|d(x,y)\}$$

Then we can write

$$v_l(x,y) = \frac{p(d(x,y)|l(x,y))\Pr\{l(x,y)\}}{\sum_k p(d(x,y)|k(x,y))\Pr\{k(x,y)\}}. \qquad (3)$$

Therefore the propagation speed depends on the likelihood ratios and on the *a priori* probabilities. The likelihood ratios can be evaluated according to assumptions on the data, and the *a priori* probabilities could be estimated, either globally or locally, or assumed all equal.

In the case of a decision between the "changed" and the "unchanged" labels according to the assumption of Laplacian distributions, the likelihood ratios are exponential functions of the absolute value of the inter-frame difference. In a pixel-based framework the decision process is highly noisy. Moreover, the moving object might be non-rigid, its various components undergoing different movements. In regions of uniform intensity the frame difference could be small, while the object is moving. The memory of the "changed" area of the previous frames should be used in the definition of the local *a priori* probabilities used in the propagation process. According to Equations (3) and (1) the two propagation velocities could be written as follows

$$v_0(x,y) = \frac{1}{1 + \frac{Q_1(x,y;0)\lambda_1}{Q_0(x,y;0)\lambda_0} e^{(\lambda_0-\lambda_1)|d(x,y)|}}$$

and

$$v_1(x,y) = \frac{1}{1 + \frac{Q_0(x,y;1)\lambda_0}{Q_1(x,y;1)\lambda_1} e^{-(\lambda_0-\lambda_1)|d(x,y)|}},$$

where the parameters λ_0 and λ_1 have been previously estimated. We distinguish the notation of the *a priori* probabilities defined here from those given in Equation (2), because they should adapted to the conditions of propagation and to local situations. Indeed, the above velocity definition is extended in order to include the neighbourhood of the considered point

$$v_l(x,y) = \Pr\{l(x,y)|d(x,y), \hat{k}(x',y'), (x',y') \in \mathcal{N}(x,y))\},$$

where the neighbourhood may depend on the label, and may be defined on the current frame as well as on previous frames. Therefore in this case the ratio of *a priori* probabilities is adapted to the local context, as in a Markovian model. A more detailed presentation of the approach for defining and estimating these probabilities follows.

From the statistical analysis of the data's mixture distribution we have an estimation of the *a priori* probabilities of the two labels (P_0, P_1). This is an estimation and not *a priori* knowledge. However, the initially labelled points are not necessarily distributed according to the same probabilities, because the initial detection depends on the amount of motion, which could be spatially and temporally variant. We define a parameter β measuring the divergence of the two probability distributions as follows:

$$\beta = \left(\frac{\hat{P}_0 P_1}{\hat{P}_1 P_0}\right)^{\beta_0(\hat{P}_0+\hat{P}_1)},$$

where $\hat{P}_0 + \hat{P}_1 + \hat{P}_u = 1$, $\hat{P}_u$ being the percentage of unlabelled pixels. The parameter β_0 is fixed equal to 4 if the camera is not moving, and to 2 if the camera is moving. Then β will be the ratio of the *a priori* probabilities. In addition, for $v_1(x,y)$ the previous "change" map and local assignements are taken into account, and we define

$$\frac{Q_0(x,y;1)}{Q_1(x,y;1)} = \frac{e^{\theta_1-(\alpha(x,y)+n_1(x,y)-n_0(x,y))\zeta}}{\beta},$$

where $\alpha(x,y) = \ln(2\delta(x,y)-1)$, with $\delta(x,y)$ the distance of the (interior) point from the border of the "changed" area on the previous pair of frames, and $n_1(x,y)$ (resp. $n_0(x,y)$) the number of pixels in neighbourhood already labelled as "changed" (resp. "unchanged"). The parameter ζ is adopted from the Markovian nature of the label process and it can be interpreted as a potential characterizing the labels of a pair of points. Finally, the exact propagation velocity for the "unchanged" label is

$$v_0(x,y) = \frac{1}{1 + \beta\frac{\lambda_1}{\lambda_0} e^{\theta_0+(\lambda_0-\lambda_1)|d(x,y)|-n_\Delta(x,y)\zeta}} \qquad (4)$$

and for the "changed" label

$$v_1(x,y) = \frac{1}{1 + \frac{1}{\beta}\frac{\lambda_0}{\lambda_1} e^{\theta_1-(\lambda_0-\lambda_1)|d(x,y)|-(\alpha(x,y)-n_\Delta(x,y))\zeta}}, \qquad (5)$$

where $n_\Delta(x,y) = n_0(x,y) - n_1(x,y)$. In the tested implementation the parameters are set as follows: $\theta_0 = 4\zeta$ and $\theta_1 = 5\zeta + 4$.

We use the fast marching algorithm for advancing the contours towards the unlabelled space. Often in level set approaches constraints on the boundary points are introduced in order to obtain a smooth and regularised contour and so that an automatic stopping criterion for the evolution is available. Our approach differs in that the propagation speed depends on competitive region properties, which both stabilise the contour and provide automatic stopping for the advancing contours. Only the smoothness of the boundary is not guaranteed. Therefore the dependence of the propagation speed on the pixel properties alone, and not on contour curvature measures, is not a strong disadvantage here. The main advantage is the computational efficiency of the fast marching algorithm.

The proposed algorithm is a variant of the fast marching algorithm which, while retaining the properties of the original, is able to cope with multiple classes (or labels). The execution time of the new algorithm is effectively made independent of the number of existing classes by handling all the propagations in parallel and dynamically limiting the range of action for each label to the continually shrinking set of pixels for which a final decision has not yet been reached. The propagation speed may also have a different definition for each class and the speed could take into account the statistical description of the considered class.

The high-level description of the algorithm is as follows:

```
InitTValueMap()
InitTrialLists()
while (ExistTrialPixels())
  {
    pxl = FindLeastTValue()
    MarkPixelAlive(pxl)
    UpdateLabelMap(pxl)
    AddNeighborsToTrialLists(pxl)
    UpdateNeighborTValues(pxl)
  }
```

The algorithm is supplied with a label map partially filled with decisions. A map with pointers to linked lists of trial pixel candidacies is also maintained. These lists are initially empty except for sites neighbouring initial decisions. For those sites a trial pixel candidacy is added to the corresponding list for each different label of neighbouring decisions and an initial arrival time is assigned. The arrival time for the initially labelled sites is set to zero, while for all others it is set to infinity. Apart from their participation in trial lists, all trial candidacies are maintained in a common priority queue, in order to facilitate the selection of the candidacy with the smallest arrival time.

While there are still unresolved trial candidacies, the trial candidacy with the smallest arrival time is selected and turned alive. If no other alive candidacy exists for this site, its label is copied to the final label map. For each neighbour of this site a trial candidacy of the same label is added, if it does not already possess one, to its corresponding trial list. Finally, all neighbouring trial pixels of the same label update their arrival times according to the stationary level set equation

$$\| \nabla T(x,y) \| = \frac{1}{v(x,y)} \tag{6}$$

where $v(x,y)$ corresponds to the propagation speed at point (x,y) of the evolving front, while $T(x,y)$ is a map of crossing times.

While it may seem that for a given site trial pixels can exist for all different labels, in fact there can be at most four, since a trial candidacy is only introduced by a finalised decision of a neighbouring pixel. In practice trial pixels of different labels coexist only in region boundaries; therefore the average number of label candidacies per pixel is at most two. Even in the worst case, it is evident that the time and space complexity of the algorithm is independent of the number of different labels. Experiments indicate a running time no more than twice that required by the single contour fast marching algorithm.

4 MOVING OBJECT LOCALIZATION USING REGION GROWING ALGORITHM

4.1 Initialisation

The change detection stage could be used for initialisation of the moving object tracker. The objective now is to localize the boundary of the moving object. The ideal change area is the union of sites which are occupied by the object in two successive time instants

$$C(t,t+1) = O(t) \cup O(t+1), \tag{7}$$

where $O(t)$ is the set of points belonging to the moving object at time t. Let us also consider the change area

$$C(t-1,t) = O(t) \cup O(t-1). \tag{8}$$

It can easily be shown that the intersection of two successive change maps $C(t-1,t) \cap C(t,t+1)$ is equal to

$$O(t) \cup (O(t+1) \cap O(t-1)).$$

This means that the intersection of two successive change maps is a better initialisation for moving object localisation than either of them. In addition sometimes

$$(O(t+1) \cap O(t-1)) \subset O(t).$$

If this is true, then

$$C(t,t+1) \cap C(t,t-1) = O(t).$$

Of course the above described situation is an ideal one, and is a good approximation only in the case of a still camera. Thus in this case, knowing also that there are some errors in change detection and that sometimes under some assumptions the intersection of the two change maps gives the object location, we propose to initialize a region growing algorithm by this map, *i.e.*, the intersection of two successive change maps. This search will be performed in two stages: first, an area containing the object's boundary is extracted, and second, the boundary is detected. The description of these stages follows.

4.2 Extraction of the uncertainty area

The objective now is to determine the area that contains the object's boundary with extremely high confidence. Because of errors resulting from the change detection stage, and also because of the fact that the initial boundary is, in principle, placed outside the object, as shown in the previous subsection, it is necessary to find an area large enough to contain the object's boundary. This task is simplified if some knowledge about the background is available. In the absence of knowledge concerning the background, the initial boundary could be relaxed in both directions, inside and outside, with a constant speed, which may be different for the two directions. Within this area then we search for the photometric boundary.

The objective is to place the inner border on the moving object and the outer border on the background. We emphasise here that *inner* means inside the object and *outer* means outside the object. Therefore if an object contains holes the inner border corresponding to the hole includes the respective outer border, in which case the inner border is expanding and the outer border is shrinking. In any case the object contour is expected to be between them at every point and under this assumption it will be possible to determine its location by the gradient-based module described in the next subsection. Therefore, the inner border should advance rapidly for points on the background and slowly for points on the object, whereas the opposite should be happen for the outer border.

For cases in which the background can be easily described, a level set approach extracts the zone of the object's boundary. Let us suppose that the image intensity of the background could be described by a Gaussian random variable with mean μ and variance σ^2. This model could be adapted to local measurements.

The propagation speeds will be also determined by the *a posteriori* probability principle. If, as assumed, the intensity on the background points is distributed according to the Gaussian distribution, the local average value of the intensity should also follow the Gaussian distribution with the same mean value and variance proportional to σ^2. The likelihood test on the validity of this hypothesis is based on the normalised difference between the average and the mean value

$$\frac{(\bar{I}-\mu)^2}{\sigma^2}$$

where $\bar{I}$ is the average value of the intensity in a window of size 3×3 centered at the examined point. A low value means a good fit with the background. Therefore the inner border should advance more rapidly for low values of the above statistics, while the outer border should be decelerated for the same values.

On the other hand it is almost certain that the border resulting from the previous stages is located on the background. Thus the probability of being on the background is much higher than the probability of being on the object. For the outer border the speed is defined as

$$v_b = \frac{1}{1 + c_b e^{-4\frac{(\bar{I}-\mu)^2}{\sigma^2}}} \tag{9}$$

where it is considered that the variance of $\bar{I}$ is equal to $\sigma^2/8$. According to Equation (3) the constant c_b is

$$c_b = \frac{P_b}{P_o}\frac{\Delta}{\sigma\sqrt{2\pi}},$$

where P_b and P_o are the *a priori* probabilities of being on the background or on the moving object, respectively. We have assumed that in the absence of knowledge the intensity on the object is uniformly distributed in an interval whose the width is Δ (possibly equal to 255). As the initial contour is more likely located on the background, P_o is given a smaller value than P_b (typically $P_b/P_o = 3$). The outer border advances with the complementary speed

$$v_o = 1 - v_b, \tag{10}$$

using the same local variance computation.

The width of the uncertainty zone is determined by a threshold on the arrival times, which depends on the size of the detected objects and on the amount of motion and which provides the stopping criterion. At each point along the boundary the distance from a corresponding "center" point of the object is determined using a heuristic technique for fast computation. The uncertainty zone is a fixed percentage of this radius modified in order to be adapted to the motion magnitude. However, motion is not estimated, and only a global motion indicator is extracted from the comparison of the consecutive changed areas. The motion indicator is equal to the number of pixels with different labels on two consecutive "change" maps reported to the number of the detected object points.

4.3 Region growing-based object localisation

The last stage of object segmentation is carried out by a seeded region growing (SRG) algorithm which was initially proposed for static image segmentation using a homogeneity measure on the intensity function [1]. It is a sequential labelling technique, in which each step of the algorithm labels exactly one pixel, that with the lowest dissimilarity. In [4] the SRG algorithm was used for semi-automatic motion segmentation.

The segmentation result depends on the dissimilarity criterion, say $\delta(\cdot,\cdot)$. The colour features of both background and foreground are unknown in our case. In addition local inhomogeneity is possible. For these reasons we first determine the connected components already labeled, with two possible labels: background and foreground. On the boundary of all connected components we place representative points, for which we compute the locally average colour vector in the *Lab* system. The dissimilarity of the candidate for labelling and region growing point from the labelled regions is determined using this feature and the euclidean distance. After every pixel labelling the corresponding feature is up-dated. Therefore, we search for sequential spatial segmentation based on colour homogeneity, knowing that both background and foreground objects may be globally inhomogeneous, but presenting local colour similarities, sufficient for their discrimination.

For the implementation of the SRG algorithm, a list that keeps its members (pixels) ordered according to the dissimilarity criterion is used, traditionally referred to as Sequentially Sorted List (SSL). With this data structure available, the complete SRG algorithm is as follows:

S1 Label the points of the initial sets.

S2 Insert all neighbours of the initial sets into the SSL.

S3 Compute the average local colour vector for a predetermined subset of the boundary points of the initial sets.

S4 While the SSL is not empty:

- S4.1 Remove the first point y from the SSL and label it.
- S4.2 Update the colour features of the representative to which the point y was associated.
- S4.3 Test the neighbours of y and update the SSL:
  - S4.3.1 Add neighbours of y which are neither already labeled nor already in the SSL, according to their value of $\delta(\cdot,\cdot)$.
  - S4.3.2 Test for neighbours which are already in the SSL and now border on an additional set because of y's classification. These are flagged as boundary points. Furthermore, if their $\delta(\cdot,\cdot)$ is reduced, they are promoted accordingly in the SSL.

When SRG is completed, every pixel is assigned one of the two possible labels: foreground or background.

5 RESULTS AND CONCLUSION

We applied the above described algorithm to the entire COST data set. The results are given in the following web page

`http://www.csd.uoc.gr/~tziritas/cost.html`

We obtained results ranging from good to very good, depending on the image sequence. The image sequence classification was always correct. The parametric motion model was estimated with sufficient accuracy. The independent motion detection was confident in the case of camera motion. The mixture of Laplacians was accurately estimated, and the initialization of the label map was correct, except for some problems caused by shadows, reflexions and homogeneous intensity on the moving objects. The fast marching algorithm was very efficient and performant. The last stage of moving object localisation can be further improved. The modelization of local colour and texture content could be possible, leading to a more adaptive region growing, or eventually a pixel labelling procedure.

Acknowledgment: This work has been funded in part by the European IST PISTE ("Personalized Immersive Sports TV Experience") and the Greek "MPEG-4 Authoring Tools" projects.

References

[1] R. Adams and L. Bischof, "Seeded Region Growing," IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol. 16, pp. 641–647, June 1994.

[2] A. A. Alatan, *et al.*, "Image Sequence Analysis for Emerging Interactive Multimedia Services–The European COST 211 Framework," IEEE Trans. on Circuits and Systems for Video Technology. Vol. 8, pp. 802–813, Nov. 1998.

[3] R. Duda and P. Hart, Pattern Classification and Scene Analysis. New York: Willey-Interscience, 1973.

[4] I. Grinias and G. Tziritas, "A semi-automatic seeded region growing algorithm for video object localization and tracking," Signal Processing: Image Communication. 2001 (to appear).

[5] M. Kim, *et al.*, "A VOP Generation Tool: Automatic Segmentation of Moving Objects in Image Sequences Based on Spatio-Temporal Information," IEEE Trans. on Circuits and Systems for Video Technology. Vol. 9, pp. 1216–1226, Dec. 1999.

[6] P. Salembier and F. Marques, "Region-based Representations of Image and Video: Segmentation Tools for Multimedia Services," IEEE Trans. on Circuits and Systems for Video Technology. Vol. 9, pp. 1147–1169, Dec. 1999.

[7] J. Sethian, "Theory, algorithms, and applications of level set methods for propagating interfaces," Acta Numerica. pp. 309–395, 1996.

[8] E. Sifakis, C. Garcia, and G. Tziritas, "Bayesian level sets for image segmentation," Journal of Visual Communication and Image Representation. 2001 (to appear).

[9] E. Sifakis and G. Tziritas, "Moving object localisation using a multi-label fast marching algorithm," Signal Processing: Image Communication. 2001 (to appear).

[10] E. Sifakis and G. Tziritas, "Fast marching to moving object location," Proceedings of the 2nd Intern. Conf. on Scale-Space Theories in Computer Vision, Corfou, Greece, 1999, pp. .

[11] T. Sikora, "The MPEG-4 video standard verification model," IEEE Trans. on Circuits and Systems for Video Technology. Vol. 7, pp. 19–31, Feb. 1997.

SEGMENTING MOVING OBJECTS: THE Modest VIDEO OBJECT KERNEL

Andrea Cavallaro[*], *Damien Douxchamps*[**], *Touradj Ebrahimi*[*], *Benoit Macq*[**]
[*] Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland
[**] Université Catholique de Louvain, Louvain-la-Neuve, Belgium
Tel: +41 21 693 2708; fax: +41 21 693 7600
e-mail: andrea.cavallaro@epfl.ch

ABSTRACT

A system separating objects moving within a slow changing background is presented. The originality of the approach resides in two related components. First, the change detection robust to camera noise which does not require any sophisticated parametric tuning as it is based on a probabilistic method. Second, the change is detected between a video frame representing a scene at a given time, and reference that is updated continuously to take into account slow variation in the background. The system is particularly suitable for indoor and outdoor surveillance. Simulation results show that the proposed scheme performs rather well in extracting video objects, with stability and good accuracy, while being of a relatively reduced complexity.

1 INTRODUCTION

Advances in micro-processors, software design and networking have made it possible to rely on more sophisticated machines capable of performing more complex operations, based on richer information. As a consequence, image understanding and computer vision is reaching a certain maturity so as to be considered for applications in every day life.

Emerging standards such as MPEG-4 and MPEG-7 have contributed to accelerate this trend. MPEG-4, as an object-based coding algorithm, allows manipulation of audiovisual objects in a compressed video, in a similar way one interacts with physical objects in the real life. This brings enhanced functionalities and applications such as efficient very low bit rate video coding where only the objects of interest are coded. Similarly, it is possible to reconstruct a photo-realistic virtual scene by taking objects from other real scenes and rendering them together in a manner similar to special visual effects in the movie industry. Such applications are obviously possible only when objects can be detected and extracted from natural scenes, either manually, in a semi-automatic way, or even in a fully automatic fashion. MPEG-7 the emerging standard for representation of audiovisual information based on a content-based approach allows for simple to sophisticated description of such content. This enables applications such as search and filtering where information with a specific content is (or is not) of interest. Video surveillance is another typical application where content of a scene has to be examined to decide if any abnormal behaviour has occured. Abnormal can vary from simple motion of certain object, to more sophisticated patterns in their behaviour.

Segmentation is one of the fundamental problems in image processing. Although human beings and most animals perform this task in a relatively straightforward manner, years of research and developments in machine vision have not yet succeeded to match the same performance. The problem of segmentation is difficult not only because of the complexity of mechanisms involved in it, but also because it is ill posed and in this sense, no unique solution exists to segment a scene. In most situations, a priori knowledge on the nature of the problem (or its solution) is needed, often as a function of the specific application in which the segmentation tool is to be used. A segmentation process leads to a partition of an image or a video sequence into regions according to a given criterion. Many of the above-mentioned applications aim at locating moving objects in the observed scene, thus a change detector can naturally drive the segmentation in a more efficient way. Change detection analysis provides a classification of the pixels in the video sequence into one out of two classes: foreground (moving objects) and background.

To this end, we combine a change detector with a background updating technique. On one hand, the change detector is designed to precisely detect object contours and to be robust to camera noise. On the other hand, the adaptive background scheme accounts for slow environmental light changes. The combination of the two (called the Video Object Kernel) allows to automatically detect multiple moving objects in long video sequences recorded by a monocular static camera. The foreground objects identified by the Video Object Kernel are then tracked along time. A successive step tranforms the 2D tracked shapes in 3D shapes. The description of the 3D shapes is finally given to the content understanding module which derives decisions on

the observed scene. The complete system is depicted in Fig. 1.

The paper is organized as follows. Section 1 describes the Video Object Kernel. Section 2 presents the results of the extraction of foreground objects and their use in the advanced video surveillance system. Finally, in Sec. 5, we draw the conclusions.

2 THE VIDEO OBJECT KERNEL

The task of the Video Object Kernel is the identification of the areas in the video sequence corresponding to moving objects. Motion cannot be directly measured in video sequences. A related measure is the luminance intensity function and its variations in time. For this reason, a simple change detection technique consists in subtracting two images. A threshold operation is then applied on the difference image. The threshold is fixed empirically, and all pixels presenting a value larger than the threshold are considered as belonging to a moving object. The threshold has to be tuned manually according to the scene characteristics [12]. This approach is therefore not suitable for automatic applications. Various methods have been proposed in the literature to automatically extract objects [3, 7, 8, 10]. A review of these method can be found in [4].

The relationship between motion and temporal changes is not unique. Temporal changes in two successive images occur not only in the area corresponding to moving objects, but also in two additional areas referred to as uncovered background and overlap of the same object [9]. The uncovered background area does not belong to a moving object, but it is detected as temporally changed. The overlap of two successive instances of the same object is hard to be detected as changed when the object is not sufficiently textured. These problems are less critical when the temporal changes are computed between the current image and a reference frame that represents the scene background [5, 6]. For this reason, we have chosen to detect changes in the current image with respect to a reference background. In addition, in order to avoid the drawbacks of a fixed reference frame, we use an adaptive background reference frame.

2.1 Adaptive background

A reliable reference frame is fondamental for the identification of moving areas through change detection. A frame captured when no objects are present in the scene is used when short sequences are analysed. However, such a frame is not always available. In addition, a fixed background image is not suitable for long sequences. In this case, changes in the enviromental illumination lead to misdetections.

For these reasons, we use an adaptive background scheme. The scheme allows to begin the detection of moving object from any time instant in the sequence, even if foreground objects are present. In this case a short set-up time is necessary to create the reference image.

In the Video Object Kernel, the computation of the adaptive background frame is an iterative process that refreshes, at an instant n+1, the background obtained from n previous frames of the sequence with the incoming n+1 frame. The background updating method uses a blending formula that weights pixels of the incoming frame, according to their chances to belong to the background. This is achieved by computing an error map. The error map takes into account both changes with respect to the previously computed background and with respect to the previous frame. A detailed description of this adaptive process is given in [11]. The block diagram describing the background updating module is depicted in Fig. 2.

This adaptive refreshment of the background brings two main advantages. First, it allows the change detection algorithm to rely on an effective reference frame even if a frame without foreground objects is not available. Second, it increases significantly the robustness to slow changes in the environmental and illumination conditions (e.g. clouds occluding the sun light or sunsets). For long sequences, indeed, when considering the first frame as reference, changes in daylight are detected as structural changes.

From a computational point of view, it is important to note that all the frames of the video sequence do not need to be used in this process. The refreshment rate is independent from the video frame rate and it can be set according to the application and the available hardware.

2.2 Change detector

The reference background frame computed and updated as described in the previous section is given as input to the change detector. The second input is the current frame of the sequence under analysis. The goal is the detection of moving objects. Since moving objects generate changes in the image intensity, motion detection is related to temporal change detection. However, besides the perturbation in the temporal changes introduced by a moving object, camera noise also heavily influences the results of the segmentation. In fact, a large number of pixels that do not correspond to a change in the real world appear as changed in the sequence due to the noise introduced by the acquisition process. To discount the effect of noise, simple change detection techniques perform a threshold operation on the difference image. The threshold is fixed empirically. All pixels presenting a difference larger than the threshold are considered as belonging to a moving object. This approach performs well only on sequences where moving objects are highly contrasted. However, thresholds have to be tuned manually according to the sequence properties. In addition, thresholds need an update along the sequence itself. These major drawbacks limit this approach for a fully automatic application.

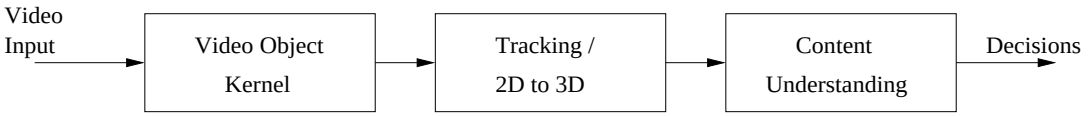


Figure 1: Block diagram of the MODEST video surveillance system. The foreground objects extracted by the Video Object Kernel are first tracked and then transformed in 3D shapes. Finally, a content understanding module derives decisions from the 3D description.

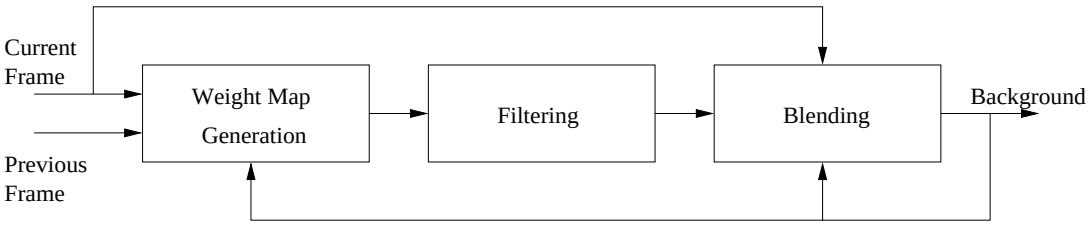


Figure 2: Block diagram of the iterative background adaptation. A background frame is generated by integrating information from the previous frame of the sequence and the already computed background.

To overcome these problems and to obtain a more flexible procedure, we adopt a method that models the noise statistics. The method is based on a statistical decision rule. According to this model proposed by Aach [1], it is possible to assess what is the probability that the value at a given position in the image difference is due to noise instead of other causes. This procedure is based on the hypothesis that the additive noise affecting each image of the sequence respects a Gaussian distribution. It is also assumed that there is no correlation between the noise affecting successive frames of the sequence. These hypotheses are sufficiently realistic and extensively used in literature. The results of the change detector is a classification of the image pixels into two groups: changed, and not changed. The classification is performed according to a significance test, after windowing the difference image. The dimension of the window can be chosen according to the application. The method and the parameter selection strategy are described in details in [4]. It is worth to notice that the only parameter whose value needs to be defined is a significance level. This is a stable parameter that is not dependent on the sequence, but on the error rate that it is tolerate for the application. This method does not require therefore any manual threshold tuning and does not severely increase the computational load compared to simple threshold techniques. An implementation of the method on a Pentium II, 300MHz processor, performs close to real time (6 frames per second, CIF format).

Since the change detector does not necessarely provide close contours, a hole filling procedure is added at the end of the scheme (Fig. 3). The results of the Video Object Kernel are then passed to the tracking module. The 2D object shapes are then translated into 3D, and their description is finally used by the content understanding module. These modules are described in the following section.

3 THE Modest SURVEILLANCE SYSTEM

The segmentation and background adaptation techniques implemented in the Video Object Kernel are integrated within the MODEST surveillance platform. Their cooperation provided satisfactory results at a reasonable computational cost. More details about the MODEST system can be found in [2]. The architecture of this system is described on Fig. 4. The sensors used are digital cameras overlooking the surveilled scene. Although several cameras are used, their field of view do not overlap and the traffic is thus analysed at a number of sparse areas. This is typical to commonly installed video surveillance systems and allows the MODEST platform to be installed without excessive hardware investments. In order to further cut investments and enhance performance, the MODEST Video Object Kernel isq designed to be placed close to the camera, leaving the scene descriptors as sole output on an IP network. This contrasts with current system that often require optical fibre to convey multiple video streams.

Besides the lower data rate at the segmention output, a first higher-level semantic information is available: the idea of object, defined at this level as an area of connected and segmented image pixels.

Once masks of objects are generated, they are used by a 3D reconstructor. This reconstractor computes position, sizes, orientation and speed of the objects as metric values. The images of objects have thus been further reduced to a small number of values of a higher semantic level, gaining substantial signification for the content understanding platform and for the final user. The geometric representation of objects is then packed

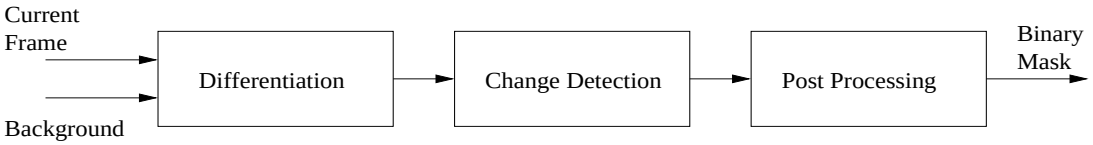


Figure 3: Block diagram of the change detector. The changes detected in the difference between a current frame and the background frame are then postprocessed to obtain masks of objects without holes.

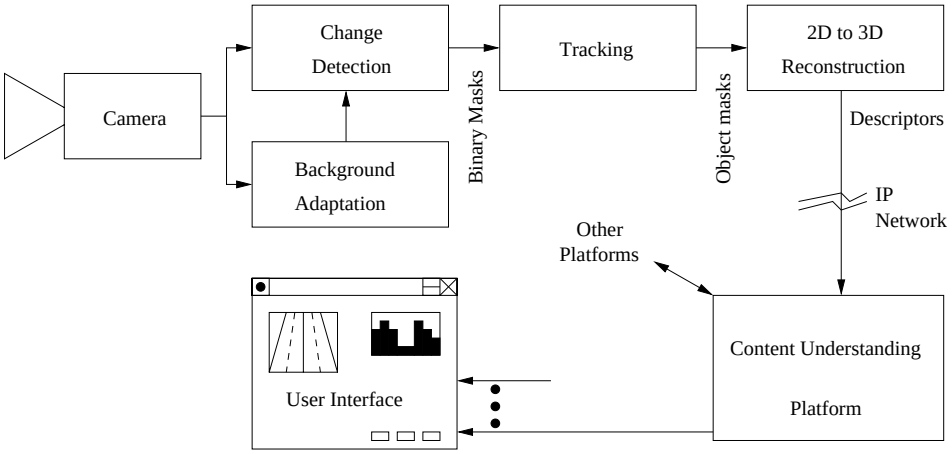


Figure 4: The architecture of the Modest surveillance system.

with other descriptors (such as color) and sent to the content understanding platform. This platform is the only part of the MODEST system to be dependent on the application. The platform is composed by a set of application-specific agents. By processing the 3D information, the agents derive statistics, analyse the behavior of the objects, and track them through the different camera sites. The data is then displayed to the user via an appropriate interface. An example is given in the next section.

4 RESULTS

The results of the Video Object Kernel are presented in this section. Since the module is addressed to both indoor and outdoor surveillance, sequences with very different characteristics have been considered. The sequences selected to present the results are the following. *Hall Monitor*, a typical example of indoor surveillance scene, from the MPEG-4 data set. *Group*, an indoor sequences characterized by many interactions and occlusions between the objects. The sequence belongs to the test set of the European IST project *art.live*. Finally, *Highway*, a typical traffic surveillance sequence from the MPEG-7 data set, is considered. The sequences contain both small and large foreground objects. The spatial resolution of the test sequences is 288×352 pixels (CIF format) and the temporal resolution is 30 images per second for *Hall Monitor* and 25 images per second for *Group* and *Highway*.

The same set of parameters has been used for all the sequences. The background refresh rate has been selected as half the original sequence frame rate.

Figure 5 presents the input and the output of the Video Object Kernel for the three sequences considered in this section. The results show a correct extraction of the foreground for both small and large objects. In addition the contours of the extracted objects are correctly defined and they are stable over time.

It is important to stress that all the sequences have been processed without changing the parameters of the Video Object Kernel. The obtained results demonstrate that the performance of the proposed method does not vary if the scene content changes. However, the results shown in this section differ from the ones of an ideal object extractor for two aspects. The first aspect is the low-pass filter effect introduced by the windowing in the change detector. The extracted contours are slightly larger then the real ones. This error is acceptable for surveillance applications. It could be corrected by a postprocessing module, if another application requires contours exactly fitting the objects. The second deviation from an ideal extraction is the presence of shadows in the change detection mask. Shadows are in fact detected as moving objects since they possess the same characteristics. The 2D to 3D conversion module in the MODEST system takes care of this problem and provides a correct description of the 3D shapes. An example of 3D shapes and object tracking is given in Fig. 6.

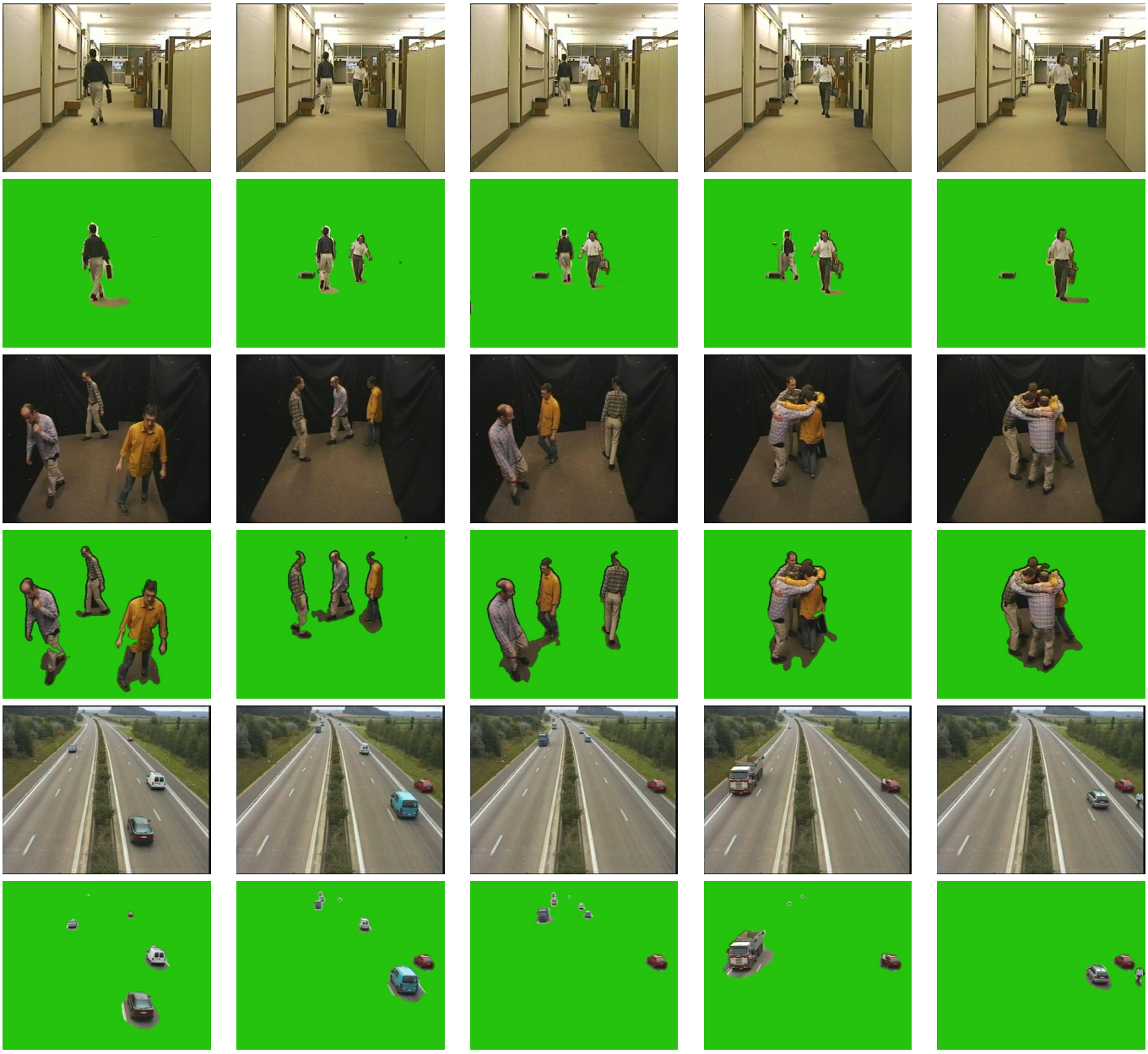


Figure 5: Original frames (first row:*Hall Monitor*, third row:*Group*, fifth row:*Highway*), and corresponding results (second, fourth, and sixth row) of the Video Object Kernel. The extraction of the video objects corresponding to the above original frame is visualized by superposing the resulting change detection mask over the original sequence. The complete sequences are available at http://ltswww.epfl.ch/~andrea/vok.html .

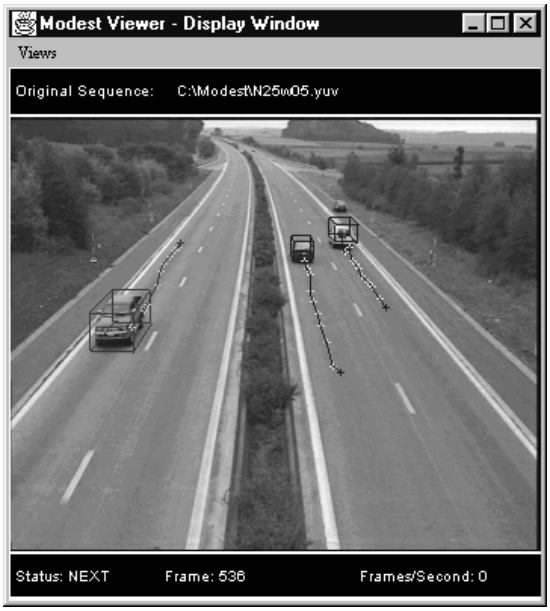


Figure 6: Example of final result of the MODEST system. The moving objects are tracked and their 3D shapes are described.

5 CONCLUSIONS

We presented here a novel scheme for extraction of video objects in a scene generated from a single camera without any specific calibration. The scheme makes use of a statistical change detection where the only parameter to set is related to a probability of detection of a moving objects (structural change) versus that of the noise of the camera. Making use of a varying reference image, which approximates scenes background, further enhances this change detection. The update of the reference allows for a better adaptation of the scheme to slowly changing backgrounds, such as outdoor scenes where changes in illumination occur.

Experimental results on long sequences show that the proposed method provides stable results in terms of detection of moving objects and accuracy of their contours. This method is a component technology to be placed in a larger framework (e.g. object-based indexing and retrieval), and has been successfully applied within an advanced video surveillance system (European project ACTS304 Modest). In this application, the information extracted from the moving objects (such as motion, shape, colour) are provided to a content understanding module which is responsible for interpreting the monitored scene.

ACKNOWLEDGMENTS

The authors whish to thank the MODEST team (http://www.tele.ucl.ac.be/MODEST/) at Laboratoires d'Electronique Philips (LEP), Instituto Superior de Ciencias do Trabalho e da Empresa (ADETTI), EPFL, and UCL. A special thanks to Benoit Mory for his work on background adaptation.

References

[1] T.Aach, A.Kaup, and R.Mester. "Statistical model-based change detection in moving video" *Signal Processing*, 31:165–180, 1993.

[2] B. Abreu, L. Botelho, A. Cavallaro, et al., "Video-Based Multi-Agent Traffic Surveillance System", Proc. of IEEE Intelligent Vehicles Symposium (IV2000), Detroit (USA), pp. 457-462, 3-5 October 2000.

[3] D. Aubert, "Passengers Queue Measurement", In *Proc. of 10th International Conference on Image Analysis and Processing*, Venice (Italy), pp. 1132–1135, 27-29 September 1999.

[4] A. Cavallaro and T. Ebrahimi, "Video Object Extraction based on Adaptive Background and Statistical Change Detection", Proc. of SPIE Electronic Imaging 2001 - Visual Communications and Image Processing, San Jose' (California, USA), pp. 465-475, 21-26 January 2001

[5] G.W. Donohoe, D.R. Hush, and N.Ahmed. "Change detection for target detection and classification in video sequences" In *IEEE Proceedings of ICASSP*, pp. 1084–1087, New-York, 1988.

[6] K.P. Karmann, A.Brandt, and R.Gerl. "Moving object segmentation based on adaptive reference images" In *Proc. 5th European Signal Processing Conf.*, pp. 951–954, Barcelona, 1992.

[7] A.Makarov. "Comparison of background extraction based intrusion detection algorithms" In *Proc. of IEEE International Conference on Image Processing (ICIP)*, pages 521–524, 1996.

[8] X. Marichal "On-line Web Application using Image Segmentation' ', In *Proc. of WIAMIS99*, Berlin, pp. 141–144, 1999.

[9] A.Mitiche and P.Bouthemy. "Computation and analysis of image motion: A synopsis of current problems and methods" *International Journal of Computer Vision*, 19(1):29–55, 1996.

[10] T. Nakanishi and K. Ishii. "Automatic vehicle image extraction based on spatio-temporal image analysis" In *Proc. of 11th International Conference on Pattern Recognition (ICPR)*, pp. 500–504, 1992.

[11] P. Piscaglia, A. Cavallaro, M. Bonnet and D. Douxchamps,"High Level Descriptors of Video Surveillance Sequences", In *Proc. of 4th European Conference on Multimedia Applications, Services and Techniques (ECMAST'99)*, Madrid (Spain), pp. 316-331, 26-28 May 1999.

[12] P. L. Rosin, "Thresholding for change detection", In *Proc. of International Conference of Computer Vision (ICCV-98)*, pp. 274–279, 1998.

SEMI-AUTOMATIC VIDEO OBJECT SEGMENTATION USING RECURSIVE SHORTEST SPANNING TREE AND BINARY PARTITION TREE

Saman Cooray[1], Noel O'Connor[2], Sean Marlow[2], Noel Murphy[2], Thomas Curran[2]

[1]Teltec Ireland, Dublin City University, Dublin 9, Ireland
[2]School of Electronic Engineering, Dublin City University, Dublin 9, Ireland
saman.cooray@teltec.dcu.ie

ABSTRACT

The objective of our work was to develop a fast and efficient tool for video content browsing and semantic video object extraction. The tool was developed using the Recursive Shortest Spanning Tree (RSST) algorithm and the Binary Partition Tree (BPT) technique. We first create an initial partition using the RSST algorithm which allows the user to specify the initial number of regions. We then progressively merge these regions to create the BPT thereby allowing the user to browse the content in a hierarchical manner. This merging step creates the binary tree with nearly double the user-specified number of homogenous regions. User interaction then allows grouping particular regions into objects. This user interaction is designed to allow object segmentation to be performed in a user-friendly manner. Any "interesting" regions can be marked in order to force them not to be further subdivided in the browsing process, which very importantly allows a small number of homogenous regions to be selected for an object. Other functionalities such as manually correcting the automatically generated results and multiple object segmentation, etc. are supported.

1. INTRODUCTION

Image content analysis and specifically object segmentation is a very important pre-processing step for emerging standards such as ISO MPEG-4 and ISO MPEG-7. In the context of MPEG-4, video object segmentation and tracking has come under extensive research in recent years. Typically, the first stage of any tracking algorithm is object segmentation performed on the first image of a sequence. The accuracy of this segmentation can determine the success or failure of the whole tracking process. The object segmentation in the first image can be performed in automatic or semi-automatic manner (i.e. areas/regions of interest defined by the user) [1].

A "semantic object" can be described as any meaningful entity in the real world with which the user may wish to interact i.e. car, television, window, ball, hair, torso, etc.. In image analysis, these video objects can be comprised of one or more automatically segmented regions. In most cases, they are comprised of multiple regions. In this context, we define a region as a homogenous area (one or multiple pixels) according to a pre-defined quantitative criterion. In general, this criterion can be any of grey level, color, texture, motion or any of the combinations [2,3,4]. In our approach, we use spatial "color" information as the only homogeneity information since the object segmentation is currently only performed on still images. However, this work could be extended to use both spatial domain and time domain information for video sequence segmentation.

The generation of objects can be fully-automatic or semi-automatic. However, fully-automatic approaches still require further research due to the fact that general video sources cannot be modeled accurately to extract semantic objects [5].

We have chosen two known techniques, RSST and BPT, to develop our tool. However, other automatic segmentation algorithms such as morphological watershed as used in [2], Pyramidal Region growing, Color Clustering could be used in place of RSST. We chose the RSST as the automatic segmentation algorithm due to its simplicity and efficiency among others [6]. The BPT method provides the hierarchical region browsing feature to the user thereby allowing the object segmentation to happen in a user-friendly manner.

As both these techniques are so-called "bottom-up" segmentation by region merging it is worthwhile to mention three notions on merging algorithms. These are merging order, merging criterion and region model [7]. The merging order defines the order in which the region links should be processed to determine the sequence of merging and it is a function of two candidate regions to be merged. The merging criterion decides whether the two candidate regions should be merged or not whereas the region model defines how to represent the resulted region.

The organization of this paper is as follows. Section 2 gives a short description of the RSST algorithm and section 3 describes the BPT technique. The implementation of our approach is then described in section 4. Section 5 and 6 then follow with results and some conclusions.

2. THE RSST METHOD

Due to its simplicity and efficiency the RSST algorithm can be considered as a very useful automatic algorithm for image segmentation. The RSST itself is a hierarchical algorithm in the sense that segmentation

starts from the finest level (i.e. single pixel level) to coarsest level (i.e. a user-specified level). For this reason, the final number of regions has to be externally specified by the user thus fixing the merging criterion to that given value. Once it reaches the given value it terminates the region growing process thus resulting a partition with the user-specified number of regions. It should be noted that though the original RSST algorithm uses the number of regions as the merging criterion introducing Peak Signal to Noise Ratio (PSNR) criterion, is also straightforward but at the expense of excessive mathematical calculations. The original algorithm is explained in [8].

Initially, each pixel in a 2D image is considered to be a region and is mapped onto a node of a graph thereby creating a set of nodes and this set contains a number of regions which is equal to the number of pixels in the image. Each node represents a region and a link is created using 4-adjacent regions initially. For each link a link-cost or a distance measure between the two corresponding regions is calculated using its luminance, chrominance and area information according to the equation given below thereby defining the merging order.

$$d(R_i,R_j) = \{[Y(R_i) - Y(R_j)]^2 + [U(R_i) - U(R_j)]^2 + [V(R_i) - V(R_j)]^2\} \times \frac{N(R_i) \times N(R_j)}{N(R_i) + N(R_j)}$$

where R_i and R_j are two candidate regions and Y(R), U(R), V(R) represent their luminance and chrominance values. N(R) represents the number of pixels in a region.

The two regions that correspond to the lowest link-cost are merged first. The color and area information of the newly formed region are calculated and the new region is represented using mean color values (Y,U and V) and sum of number of pixels of the two regions. This process removes the above link from the graph thereby constructing a spanning-tree of the initial graph. Due to this spanning, the affected links are updated with new region nodes and hence new link-costs. Repeating the same process reduces the number of regions to the user specified value. In this algorithm the mean color is used as the region model.

3. THE BPT METHOD

It has been proposed that a wide range of applications such as filtering, segmentation, information retrieval and visual browsing can be supported using the BPT technique [9]. Since arbitrary shaped regions are represented using this binary tree, visual content searching and browsing can be made fast and also due to its simplicity this technique can be efficiently utilized for interactive multimedia applications. The tree describes the regions and their spatial relationships within the scene.

The BPT creation process starts from a given initial partition. The regions belonging to the initial partition are represented in the leaves of the tree. The rest of the remaining nodes of the tree correspond to the regions created by the merging process. The merging of two regions at a time is done according to a defined merging order while maintaining the "Father" and "Children" nodes relationship. By keeping track of merging order of each region a final tree is created and each region is assigned a level thereby facilitating a hierarchical representation of the original image. The merging criterion in BPT always remains fixed and the merging of regions continues until one single region is obtained, i.e. the root node, leaving the total number of regions in the tree equal to only one less than double the initial number of regions. The binary tree with this set of regions represents the image at different scales of resolution. In order to create the BPT, a color homogeneity criterion or both color and motion homogeneity criteria can be used [9]. Fig. 1 shows a simple example of a BPT with 35 regions and 4 levels for an initial partition of 18 regions. The black nodes correspond to the regions in the initial partition whereas the grey color nodes represent the merged regions. Also, the numbers shown on the right are different partition levels of the tree which are used to represent the image in different partition levels.

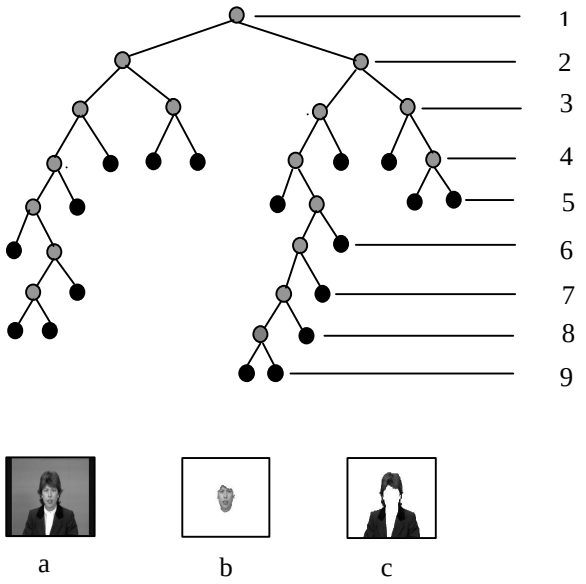


Fig. 1. A BPT with 9 levels (a) original image (b) a region belonging to level 4 (c) a region belonging to level 3

4. EXPERIMENTAL WORK

Our approach can be divided into two parts. Firstly, a set of homogenous regions is created automatically using RSST and represented hierarchically using BPT. Secondly, any interesting regions are selected manually to group them into objects. The first part consists of running both RSST and BPT for a given number of regions which is supplied by the user. This parameter is provided to the RSST algorithm through a Graphical User Interface (GUI). The input to the BPT is provided from the RSST algorithm to create the binary tree. We use the same merging order and region model criteria for both the techniques.

When creating the binary tree region, parameters such as area and color are calculated and they are attached to each node of the tree. Geometry descriptors such as shape, size, position and rotation are extremely useful for information retrieval applications and could be calculated and attached to these tree nodes, however, such methods are not used within this work as there is no need to use them in this context of object segmentation. A screen-shot of our GUI is shown in Fig. 2.

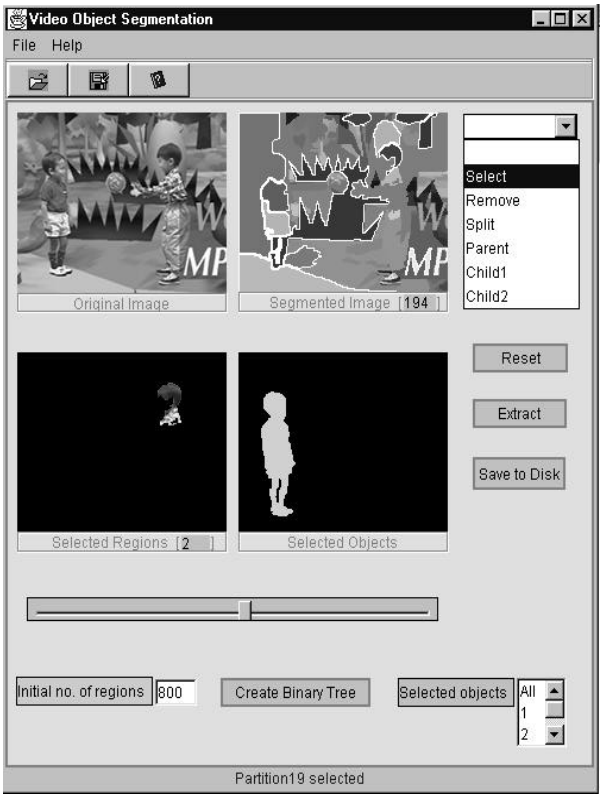


Fig. 2. A screen-shot of the GUI

The browsing of different levels of the partition tree is provided using a slider in this GUI. The number of levels in the tree determines the number of scale of resolutions the image can be browsed. However, in our approach, the user can mark any “interesting” regions using a mouse click to force them not to be subdivided further while browsing the tree. It is mainly this process which makes the object selection process easier and faster in addition to the feature of hierarchical representation provided by BPT. Further mouse clicks facilitate the options to “select”, “remove” “split”, “show parent”, “show child1” and “show child2” of the selected region. Having selected the regions which should compose the object the user can select “EXTRACT” to end the object selection process. The extracted objects are shown in a list allowing the user to select any of the object/objects to get the final segmentation thereby facilitating multiple object segmentation.

We implemented this tool entirely in Java language using the Borland JBuilder Integrated Development Environment under Windows platform. Therefore, we can port this tool easily to any other platforms and also integration of this tool into any other application should not be too difficult.

5. RESULTS

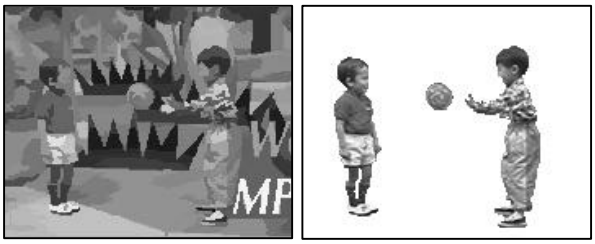
The results we obtained for three images from the MPEG-4 test sequences “Table Tennis”, “Children” and "Foreman" are shown in Fig.3, Fig. 4 and Fig. 5 respectively. Note that the initial partitions are displayed with mean grey and the extracted objects are shown in original grey color. As the most interesting feature in our approach is the user interactivity to reduce the complexity of object selection process we present the results in terms of number of mouse clicks taken for selecting the target object. The person playing table tennis was selected as the semantic object in the first experiment. In order to generate fine regions, the RSST algorithm was run for 700 regions which was the initial partition to the BPT. The binary tree contained 1399 homogeneous regions and 26 levels. The results showed that only 13 mouse clicks were required to extract the above said object which is shown in Fig.3b. The initial partition is shown in Fig. 3a.

In the second experiment, three semantic objects were selected where two of them were the two children playing with the ball in the picture. For this experiment, an initial partition of 800 regions was created from the RSST and therefore the BPT provided 1599 homogenous regions with 34 levels. To extract the first object which is the child on the left, it required 15 mouse clicks. The next child on the right required 16 mouse clicks and the “ball” required only one mouse click. Fig. 4a shows the initial partition of 800 regions and fig. 4b shows the results of three semantic objects extracted for multiple object segmentation.

In our third experiment, "Foreman" person was selected as the semantic object. The initial partition was created for 1600 regions in order to obtain finer regions and therefore a total number of 3199 homogeneous regions with 36 levels appeared in the binary tree. In order to extract this object, it required 32 mouse clicks. It clearly shows that this particular object extraction from this image is more difficult than in the first two experiments. The initial partition and the extracted semantic object are shown in Fig. 5a and Fig. 5b respectively.

a b

Fig. 3. Segmentation results for Table Tennis sequence (a) initial partition of 700 regions (b) extracted semantic object



a b

Fig. 4. Segmentation results for Children sequence (a) initial partition of 800 regions (b) extracted semantic objects

a b

Fig. 5. Segmentation results for Foreman sequence (a) initial partition of 1600 regions (b) extracted semantic object

6. CONCLUSION

The main objective of our work was the development of a tool to facilitate semantic video object segmentation from still images using a combination of a conventional segmentation algorithm (namely the RSST) and an efficient image representation approach (namely the BPT). In this paper, we have discussed how semantic video objects with different complexities can be extracted by combining the above two techniques and adding further functionalities.

The creation of the initial partition is computationally efficient due to the nature of RSST. Furthermore, since the number of regions in the partition is specified by the user it is straightforward to generate an initial partition of the required granularity. The BPT is a relatively simple approach to segmentation representation, however, introducing it into this work makes the region browsing process fast and efficient.

The results we have obtained are quite promising and illustrate the potential usefulness of this tool in the context of future MPEG-4 and MPEG-7 applications. For example, this tool (combined with a suitable tracking step) could be used to facilitate segmentation of semantic objects in video sequences in order to create Video Object Planes (VOPs) for subsequent MPEG-4 encoding. Similarly, by incorporating geometry descriptors and color features this tool could be used for future MPEG-7 applications. Currently we are extending this work towards automatic video object tracking.

REFERENCES

[1] Ferran Marques, Beatriz Margotegui, Ferran Meyer, "Tracking areas of interest for content-based functionalities in segmentation-based video schemes," IEEE International Conference on Accoustic, Speech and Signal Processing, vol. 2, pp. 1224-1227, May 1996.

[2] B. Marcotegui, P. Correia, R. Mech, R. Rosa, M. Wollborn and F. Zanoguera, " VOGUE: The MoMuSys video object generator with user environment," Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'99), June, 1999.

[3] P. Salembier and F. Marques, "Region-based representations of image and video: Segmentation tools for multimedia services," IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 8, pp. 1147-1169, December 1999.

[4] Roberto Castagno, Touradj Ebrahimi and Murat Kunt, "Video segmentation based on multiple features for interactive multimedia applications," IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 5, pp. 562-571, September 1998.

[5] Seungyup Peak, Ana B. Benitez and Shih-Fu Chang, "Self-Describing schemes for Interoperable MPEG-7 multimedia content descriptions," Symposium on Electronic Imaging: Visual Communication and Image Processing, IST/SPIE, January 1999.

[6] J. Mulroy, "Video content extraction: Review of current automatic segmentation algorithms," Workshop on Image Analysis for Multimedia Interactive Services 1997 (WIAMIS'97), June 1997.

[7] Luis Garrido and Philippe Salembier, "Region based analysis of video sequences with a general merging algorithm," In IX European Signal Processing Conference (EUSIPCO), vol. 3, pp. 1693-1696, September 1998.

[8] O. J. Morris, M. J. Lee and A.G. Constantinides," Graph theory for image analysis :an approach based on the shortest spanning tree," IEE Proceedings, vol. 133, no. 2, pp. 146-152, April 1986.

[9] Philippe Salembier and Luis Garrido, "Binary partition tree as an efficient representation for image processing, segmentation and information retrieval," IEEE Transactions on Image Processing, vol. 9, no. 4, pp. 561-576, April 2000.

A SEGMENTATION TECHNIQUE BASED ON MERGING OF COLOUR AND MOTION INFORMATION

P. Caneva, L. Capodiferro**, A. Pettorossi**

*INFOCOM Dpt., University of Rome "La Sapienza", via Eudossiana 18, 00184 Rome, Italy
**Fondazione Ugo Bordoni, via B. Castiglione 59, 00142 Rome, Italy
Tel.: +39 06 54802132; Fax: +39 06 54804405; email: licia@fub.it

ABSTRACT

In this contribution a technique for video segmentation based on joint colour and motion information is presented. The technique entails pre- and post-processing stages in addition to clustering algorithms to define homogeneous colour patches. As far as motion detection is concerned, enhanced robustness is achieved by extended depth analysis and motion estimation reliability assessment. Final merging of colour and motion information is made by classifying the global state of motion of each colour patch.

keywords: video segmentation, wavelet decomposition, colour histogram, region growing, motion analysis, reliability.

1. INTRODUCTION

The scope of this work is to describe a reliable technique for isolating moving objects from background in video sequences, to be employed in various multimedia applications, including object based coding procedures (such as MPEG 4 and MPEG 7) and image analysis.

Most conventional segmentation techniques are based on motion analysis. Maps of motion assigned to each pixel, extracted by comparison of consecutive frames, are used to separate moving objects from scene background with various criteria. However, motion analysis presents serious robustness problems, related to a variety of causes. Often, the represented patterns present inherent ambiguity which results in unreliable maps. In addition, sudden geometrical and luminance modifications produce motion artefacts. More essentially, motion analysis is insufficient for segmentation in many situations where scenes are more or less "static". As a result, motion based segmentation appears very poor in comparison with the outstanding capability possessed by the natural systems for interpreting the scenes and for recognising different objects. Today, insufficient knowledge of high level natural vision mechanisms prevents us to emulate them, and to attain comparable effectiveness. Nevertheless, it is possible to further exploit low level (non syntactic nor semantic) information contained in the observed images about the represented objects, beyond motion information. For this reason, we present here a technique for merging together motion and colour information for image segmentation [1].

The block diagram of the proposed segmentation technique is shown in fig.1. In our approach, colour segmentation is performed in the Y, Cr, Cb domain. It has been experienced that a *colour pre-processing* stage of the video signals is always recommended to obtain sufficiently well defined regions. This has leaded us to operate a "simplification" of the original image before colour segmentation. To this purpose, a non-linear processing in the wavelet domain has been applied. It consists of suppressing small valued wavelet coefficients (shrinking) of both luminance and chrominance components.

After simplification, the colour histogram of the image is calculated. This histogram serves to initiate region growing, which allows to separate, in each video image, different homogenous colour regions (patches).

A *colour post-processing* step follows, necessary to reduce smudging of some colour regions. This is based on edge information.

As far as motion based segmentation is concerned, the motion detection is based on a quad-tree block matching algorithm that estimates the dense motion field and also evaluates the "reliability" associated with each detected motion vector. Techniques for isolating camera motion are also employed.

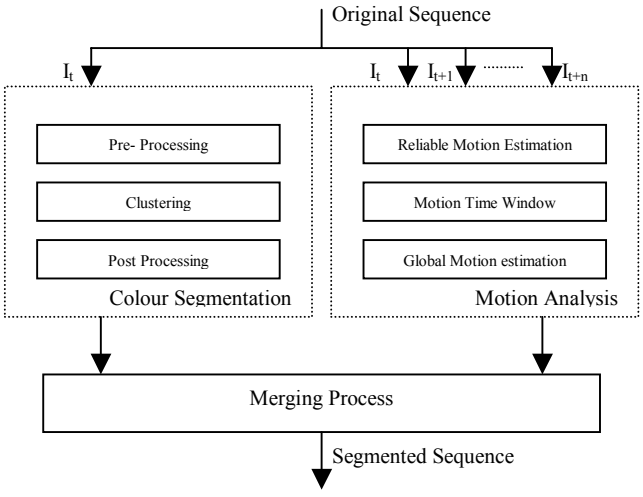


Figure 1. Block diagram of the segmentation technique

The final merging of colour-based and motion-based segmentation is accomplished by assigning a unique motion vector to each colour patch, assuming that consecutive frames of the video sequences are so close that patches can be considered as rigid patterns subject to a pure translational motion.

In the following, we first illustrate in Section 2 the colour segmentation procedure. Then, in Section 3, the motion analysis is presented in some details, and finally the merging strategy is described in Section 4.

2. COLOUR SEGMENTATION

The colour segmentation procedure, performed in the Y, Cr, Cb colour space, consists of three basic steps: pre-processing, clustering and post-processing.

As we said, the pre-processing operation is recommended to attain a simplified representation of the video images, where unessential details are eliminated. This is obtained by separate operation on luminance and chrominance components [2].

The luminance simplification is performed in the wavelet domain, as schematised in the block diagram shown in fig.2.

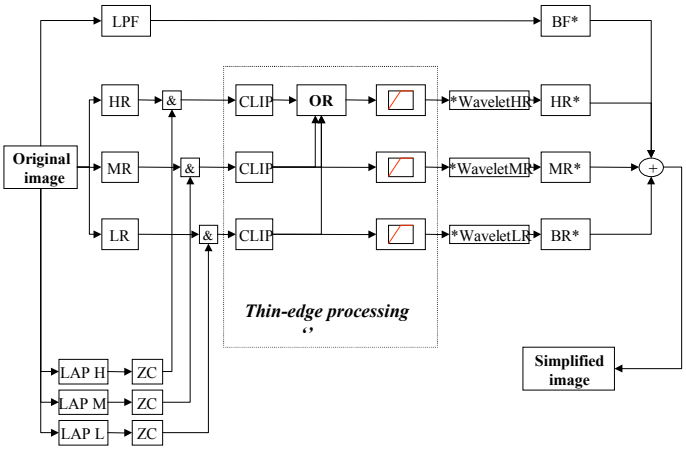


Figure 2. Pre-processor block diagram

Starting from the original image, edges at different resolution layer are extracted, using a dyadic wavelet decomposition based on a dyadic Circular Harmonic Wavelet (CHW). CHWs form a family of polar separable complex wavelets, whose azimuthal shape is complex harmonic, modulo $n2\pi$. In particular, the first order CHWs (n=1) are tuned to edges, which are represented in the wavelet domain by alignments of coefficients whose magnitude is proportional to the edge strength, and whose phase measures the edge orientation. We have employed three wavelet resolution layers (see fig. 2). Wavelet analysis performed through a law pass filter (LPF) and high resolution (HR), medium resolution (MR) and law resolution (LR) band pass filters. At the same time the images undergoes three corresponding laplacian operators (LAP H, LAP M, LAP L) followed by zero crossing (ZC) detectors. The output of ZCs are used to isolate wavelet coefficients on relative maxima of the wavelet planes.

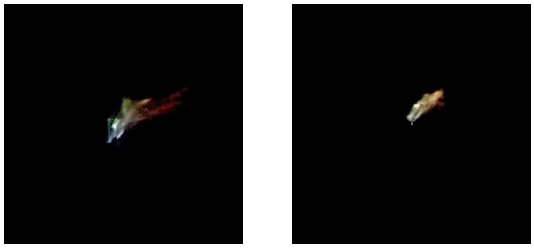
The simplification algorithm presents the structure of a typical Linear - Zero-memory non linear - Linear processing scheme. After wavelet transformation, the image undergoes a point by point non-linear transformation, and finally is reconstructed using the inverse wavelet transform process (performed by the filters BF*, HR*, MR*, BR*). The non-linear operation consists before in clipping the edge wavelet coefficients and then by passing whose value falls under a threshold, leaving unchanged the larger ones (shrinking). The result of this operation is a smeared image, lacking of minor details, but preserving the visually essential structures. This process greatly simplifies the colour segmentation operation. To give an idea of the impact of the simplification process, in fig. 3 the original image is shown together whit the luminance simplified image.

a) **b)**

Figure 3. a) original image **b)** simplified image

The colour segmentation consists of a region growing process followed by a subsequent merging adjacent patches under a similarity criterion. In order to obtain consistent frame-to frame results irrespective of image boundary changes, the region growing process is initialised with reference to maxima of the colour histogram.

For example in fig. 4, the colour histograms (Cr, Cb), for frames extracted from "container" and "mother and daughter" sequences, are shown [3].



a) **b)**

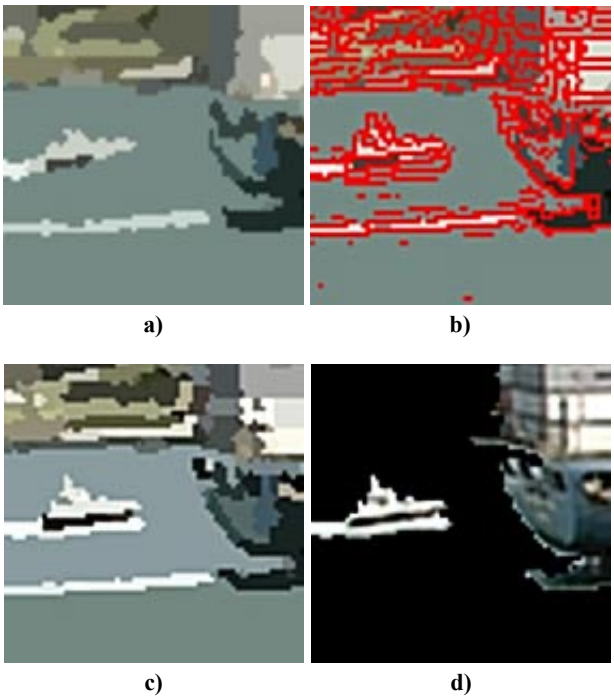
Figure 4. Colour histogram for frames extracted from: **a)** "container" sequence; **b)** "mother and daughter" sequence

Starting with “seed” pixels having the colour of the histogram maximum, region growing proceeds assigning to the same region neighbouring pixels that have similar colour properties, according to the Euclidean distance function [4]:

$$\Delta C = \sqrt{(\Delta Y)^n + (\Delta Cr)^m + (\Delta Cb)^m}$$

Where ΔY, ΔCr, ΔCb, stay for distances of luminance, and chrominance components respectively, and where the m, n exponents are chosen on the basis of the characteristics of the colour histogram. Notice that the distance used for region growing includes also the luminance component, while seed pixels are determined by chrominance histogram only. It could be observed that the "container" colour histogram is enough scattered in the Cr, Cb plane while the "mother and daughter" histogram is concentrated in a small region. This leads to different values of m, n. In the former case m=1 and n=2, in the latter m=n=2.

After this operation, the histogram is updated by removing the pixels assigned to the determined regions. The process is repeated starting with the new maximum of the histogram, and so on, until the whole image is visited by the region growing procedure.



a) **b)**

c) **d)**

Figure 5. Bordering operation: **a)** region smudging; **b)** thin edges extraction in the wavelet domain; **c)** regions bordering; **d)** final segmentation

Still, the segmentation determined by the region growing procedure presents often scattered boundaries, due to smootness of the colour transitions compared to background noise. For this reason, a edge based processing stage is applied, using edge information that had already been extracted in the wavelet domain. Colour patches are cut by the edges, so generating a larger set of patches, whose boundaries do include edges.

In particular, fine localisation of edges is performed by isolating, at the finest resolution layer, wavelet coefficients corresponding to the zero crossing of the Laplacian operator [5]. In fig. 5 a detail from a frame of "container" sequence is shown along the bordering post-processing operation.

3. MOTION ANALYSIS

The motion analysis is a separate process which is composed of three separate functions: “reliable” motion estimation, motion time window memory and global motion estimation.

3.1 Reliable Motion Estimation

At first, a quadtree blockmatching algorithm is employed to estimate the dense motion field: this allows to obtain fine details of the field in vicinity of contours. In order to measure the reliability of the motion field estimated the following quantity is calculated:

$$REL(\boldsymbol{d}) = \min_{\boldsymbol{d'}} |\, MSE(\boldsymbol{d}) - MSE(\boldsymbol{d'}) \,|$$

$$\boldsymbol{d'} = (d_1 \pm i, d_2 \pm j) \quad \text{with} \quad i = 0,1 \quad j = 0,1 \ \text{except} \ i = j = 0$$

where d is the estimated motion vector and (d_1,d_2) its coordinates.

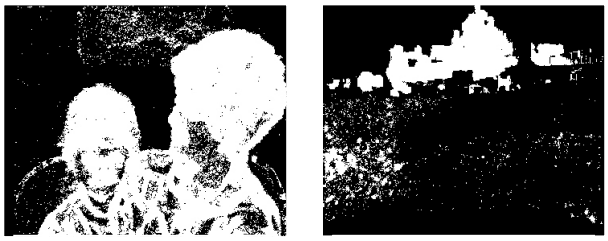
It words, it simply represents the minimum value of the matching error increment while perturbing the estimated motion vector by one pixel in the entire neighbour (with distance + or – 1). Greater the minimum increment, greater the reliability of the motion estimate.

For subsequent processing we consider only the reliable part of the motion field, i.e. the motion vectors with reliability measure greater than a given threshold, thus allowing greater robustness of the whole procedure.

3.2 Motion Time Window

For segmentation, we are interested in selecting foreground object from background even if they remain still along many frames and than began to move (this is the case for instance of the mother and daughter sequence). For this reason, we define a time window extended into the past, and overlay in the current frame (using the “or” rule) the reliable motion field from the previous frames falling in the defined window. Of course, this procedure implies a delay of delivery of the segmented sequence, but produces far more robust results.

In fig. 6, two different examples of motion time window maps are reported. The temporal extension of the window depends on the motion content of the sequence and must be chosen as a trade-off between accuracy and real time requirements.

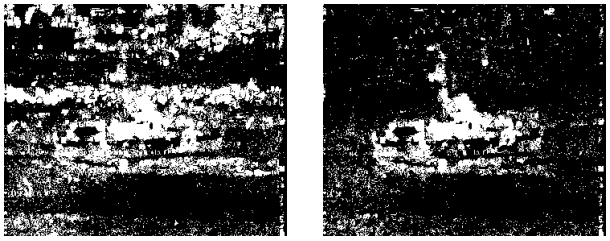


a) **b)**

Figure 6. Motion time window maps: **a)** "mother and daughter" sequence **b) "**container" sequence

3.3 Global Motion Estimation

Moreover, the egomotion of the camera is subtracted to the motion field in order to isolate the objects of interest. This is simply performed by selecting the motion parameters possessed by the majority of pixels. At the moment, no compensation for camera rotation and zooming has been implemented. An example of camera motion subtraction is reported in fig.7.



a) **b)**

Figure 7. a) reliable motion field; **b)** egomotion subtraction

The results of the three motion functions above described are then collected together as motion based segmentation results for the following merging operation.

4. MERGING OF COLOR AND MOTION INFORMATION

Colour and motion based segmentation are finally combined to yield a unique result. This merging process consists of considering each colour patch as rigid object subject to translational motion. The validity of this assumption is preserved by keeping sufficiently small the average patch dimension. For each patch, the motion field weighted by its reliability value (see step 3.1) is integrated over its area, producing a unique patch motion vector. Afterwards, adjacent patches sharing the same motion vector and with a cumulative reliability exceeding a pre-assigned threshold are grouped togheter to form the segmented zones.

This procedure takes advantage of the contour precision due to the use of the edges and of the robustness due to the integration of the motion vector weighted by estimation reliability over homogeneous colour patches. Of course, this robustness is the result of a trade-off dictated by the said necessity of keeping the colour patch areas sufficiently small. In fig.7, segmentation results obtained using the procedure illustrated above from different MPEG4 test sequences are shown.

a) **b)**

c) **d)**

Figure 8. Segmented images extracted from **a)** "mother and daughter"; **b)** "table"; **c)** "container"; **d)** "coastguard" sequences

5. REFERENCES

[1] Caneva Paola, "Segmentation techniques for colour video sequences", Degree Thesis, INFOCOM Dpt., University of Rome "La Sapienza", Rome, Italy, October 1999.

[2] Pettorossi Alessandro, "Multiresolution technique for video processing in multimedia applications", Degree Thesis, INFOCOM Dpt., University of Rome "La Sapienza", Rome, Italy, October 1999.

[3] A.R.Weeks, L.J.Sartor, H. R: Myler, "Histogram specification of 24-bit colour images in the colour difference (C-Y) colour space", Journal of electronic Imaging July 1999, Vol 8(3).

[4] N. Ikonomakis, K. Plataniotis, M. Zervakis, A. N: Venetsanopoulos, "Region Growing and Region Merging Image Segmentation", Proceedings of 13th International Conference on Digital Signal Processing, pp.229-302, 1997.

[5] L. Capodiferro, G: Andreani, S: Puledda, G. Iacovitti, "Decomposition of still and video images with edge segments", in Wavelets Appl. in Signal and Image Processing VII, Proc.SPIE'99, July 1999.

OBJECTIVE EVALUATION CRITERIA FOR 2D-SHAPE ESTIMATION RESULTS OF MOVING OBJECTS

Roland Mech
Institut fuer Theoretische Nachrichtentechnik und Informationsverarbeitung
Universität Hannover, Appelstr. 9A, 30167 Hannover, Germany
Tel: +49 511 762-5308, Fax: +49 511 762-5333
e-mail: mech@tnt.uni-hannover.de

Ferran Marqués
Universitat Politècnica de Catalunya, Campus Nord – Mòdul D5
C/ Jordi Girona 1-3, Barcelona (08034), Spain
Tel: +34 93 401 64 50, Fax: +34 93 401 64 47
e-mail: ferran@gps.tsc.upc.es

ABSTRACT

The objective evaluation of 2D-shape estimation results for moving objects in a video sequence is still an open problem. First approaches in the literature evaluate the spatial accuracy and the temporal coherency of the estimated 2D object shape. Thereby, it is not distinguished between several estimation errors located around the object contour and a few, but larger, estimation errors. Both cases would lead to a similar evaluation result, although the 2D-shapes would be visually very different. In order to overcome this problem in this paper a new evaluation approach is proposed. By this, the evaluation of the spatial accuracy and the temporal coherency is based on the mean and the standard deviation of the 2D-shape estimation errors.

1 INTRODUCTION

One major problem in the development of algorithms for 2D-shape estimation of moving objects, is to assess the quality of the estimation results. Up to now mainly subjective evaluation, i.e. tape viewing, is used in order to decide upon the quality of a certain algorithm. Although this is very helpful and gives already some indication of the resulting quality, this procedure very much depends on the subjective conditions, i.e. the attending people, the time of viewing, the used video equipment, etc.

In the literature, first approaches for objective evaluation of 2D-shape estimation results can be found: During the standardization work of ISO/MPEG-4 [6], within the core-experiment on automatic segmentation of moving objects it became necessary to compare the results of different proposed 2D-shape estimators, not only by subjective evaluation, but also by objective evaluation. The proposal for objective evaluation [9], which was agreed by the working group, uses an a-priori known 2D-shape in order to evaluate the estimation result. This 2D-shape is denoted *original* 2D-shape, and has to be created once in an appropriate way, e.g. by manual segmentation of each frame or by colour-keying. Also the usage of synthetic image sequences is thinkable, where the 2D-shape is known. The 2D-shape of a moving object can be represented by a binary mask, where a pel has *object-label* if it is inside the object and *background-label* if it is outside the object. In [9], such a mask is called *object mask*. There, two objective evaluation criteria are defined:

- The first criterion evaluates the spatial accuracy of an estimated 2D-shape. For this, the amount of pels is determined that have different labels in the estimated and the original mask. Then, this value is normalized by the size of the object, which is given by the amount of pels with object-label in the original mask.
- The most subjectively disturbing effect is the temporal incoherence of an estimated sequence of object masks. This is evaluated by the second criterion. The number of pels with opposite label between two successive frames is calculated for the original and the estimated sequence of object masks. For each frame, the difference of these two values is build and normalized by the size of the object. If the resulting value is large, this hints to a big difference in activity between the original and the estimated 2D-shape.

Beside the ISO/MPEG-4 core-experiment, this objective evaluation approach was used by the European projects COST 211 [3] and ACTS/MoMuSys [4]. However, the approach has the following shortcomings:

1. The criterion for spatial accuracy does not distinguish between a lot of small deviations between the estimated and original mask (case 1) and a few, but larger, deviations (case 2). Both cases can lead to the same value for spatial accuracy, although they are visually very different.

2. The same problem exists for the temporal coherency criterion, where it is not distinguished between a lot of small contour activities and a few, but larger, ones.

3. There is another problem of the criterion for temporal coherency evaluation in the case if the camera or the object moves between two frames. Then, changes in the object mask between these two frames are either caused by movement or by contour activity, which is not distinguished by the criterion.

Within the project COST 211 the above approach has been further developed [5][8]:

- For evaluation of the spatial accuray it is distinguished between pels that have object-label in the estimated object mask, but not in the reference mask, and vice versa, i.e. if the estimated mask is too large or too small. Furthermore, the impact of a misclassified pel on the criterion for spatial accuracy grows with its distance to the object contour. By these enhancements, the evaluation of 2D-shape estimation results can be adapted to given applications.

- For evaluating the temporal coherency two criteria are used. By the first, local instabilities are investigated by looking at the variation of the spatial accuracy criterion between successive frames. By the second, it is assumed that the 2D-shape is well estimated, but oscillates around the reference shape. For this case the distance between the gravity centers of the estimated and the original object mask is investigated for succeeding frames.

The third problem is solved by the new evaluation criteria for temporal coherency. However, the first and the second problem stated above are neither solved by this approach nor by the approach in [2], where additionally geometric features like the size and position of an object as well as the average colour within an object area are evaluated based on the estimated and the reference object mask.

In this paper a simple approach for objective evaluation of results from a 2D-shape estimation is proposed, which solves in addition to third problem also the first and the second problem. As by the approaches in the literature, the spatial accuracy and the temporal coherency of an estimated 2D-shape are evaluated by comparing it with the corresponding original 2D-shape. Thereby, the mean and the standard deviation of the 2D-shape estimation errors are determined.

The approach was developed within a collaboration between the Universitat Politècnica de Catalunya and the University of Hannover [7].

2 OBJECTIVE EVALUATION CRITERIA

2.1 Spatial Accuracy

The spatial accuracy of an estimated 2D-shape of a moving object is defined by the spatial distance between the original 2D-shape and the estimated one: For each pel i on the original object contour the distance d_i to the estimated object contour is measured. From these distance values the mean and the standard deviation are calculated, which are then normalized by the maximal diameter of the object in the original object mask, resulting in the normalized mean $\overline{d}$ and normalized standard deviation σ_d. While the mean is a measure for the average distance between the original and the estimated object contour, the standard deviation gives an idea, how different the measured distances are. The standard deviation is small if the deviation between the original and the estimated contour is quite equal for all contour pels. The standard deviation grows with the difference of the measured distance values.

For measuring the distance values between the two object contours, the following algorithm is used (Figure 1): By a first step, the original and the estimated contour are split into parts that are assigned to each other. This is done by going around the original contour and determining for each pel on the contour the perpendicular straight line. The intersection point between this straight line and the estimated contour is the corresponding pel on the estimated contour. For two succeeding pels on the original contour this leads to the corresponding part of the estimated contour, which is surrounded by the two intersection points (see zoomed area of Figure 1). In the special case that the intersection point belongs to an already assigned part of the contour or intersects the original contour first, the assignement would be unvalid, and therefore the next pel on the original contour is processed. This goes on until the resulting intersection point is not yet assigned, and the original contour is not intersected as first. The part of the estimated contour surrounded by this intersection point and the previous valid intersection point is assigned to the part of the original contour, which is surrounded by the latest processed pel and the precceeding pel for that the intersection was valid.

By the second step, the distance between measure points on the original contour and on the estimated contour is calculated. A measure point is defined as the point in the middle of two succeeding pels on the original or estimated contour. For each measure point on the original contour, the average distance to all measure points within the corresponding part of the estimated

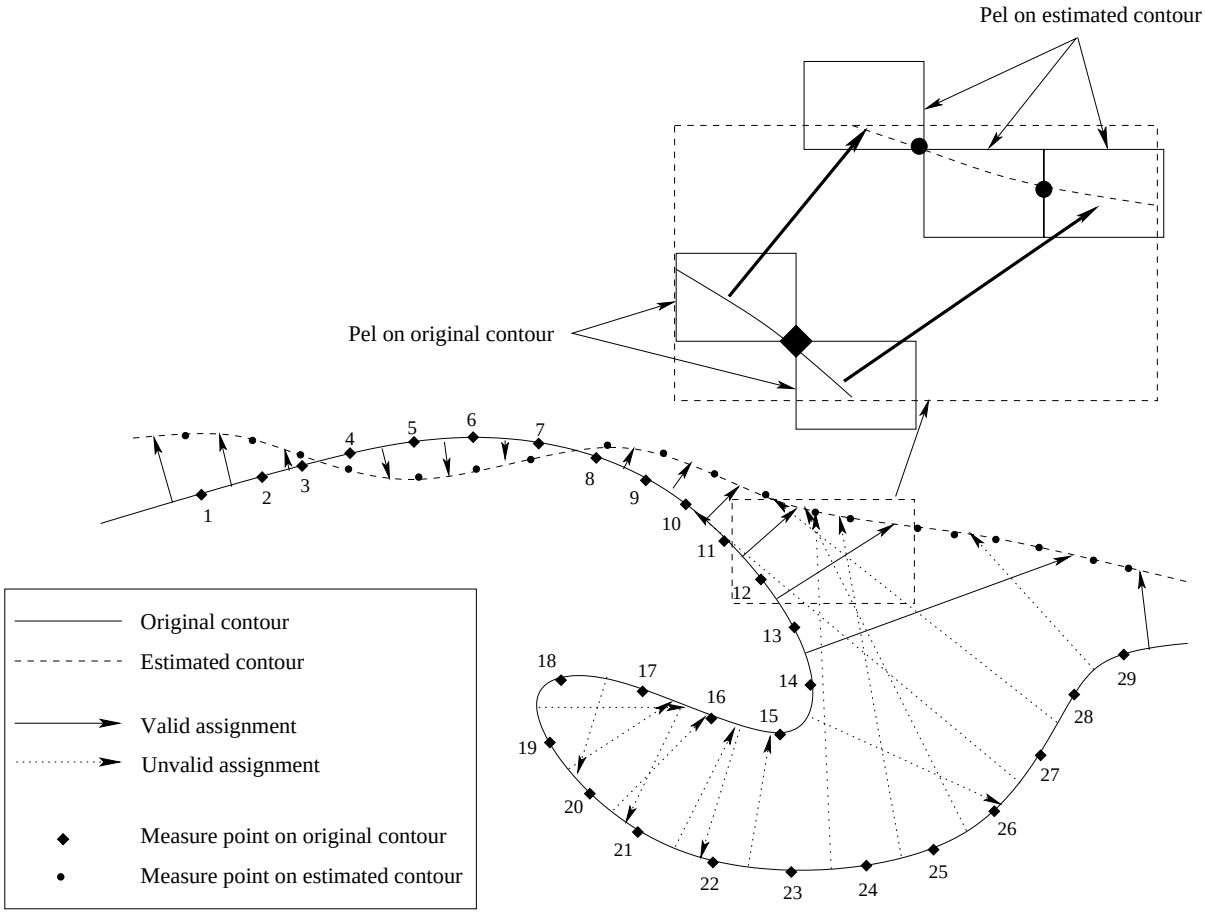


Figure 1: Determination of the spatial distance between the original and the estimated object contour.

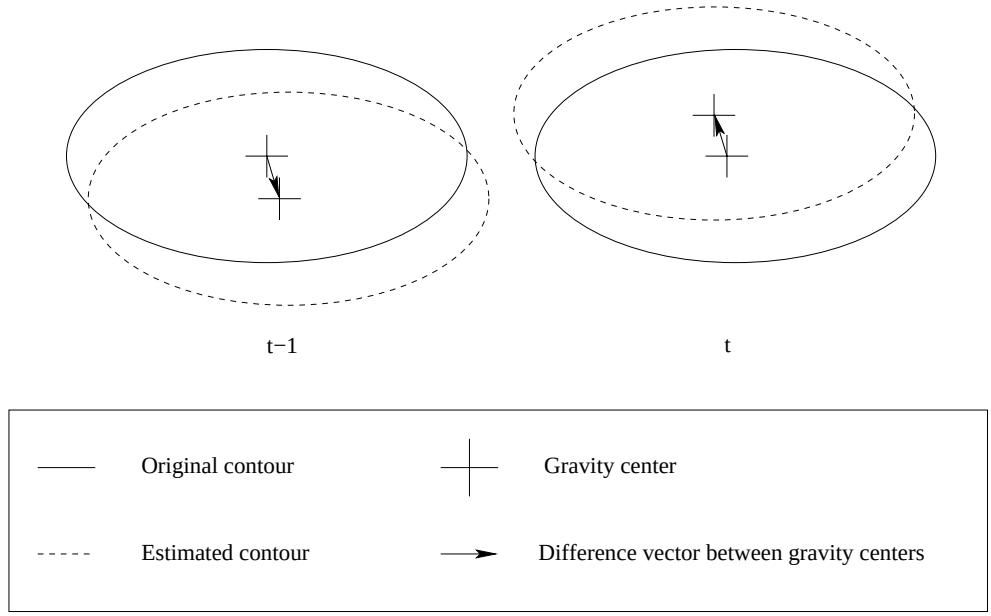


Figure 2: Evaluation of the temporal coherency by investigating the variation of the gravity centers of the original and the estimated object contour between succeeding frames at time instances $t-1$ and t.

contour is calculated. In the example, which is shown in the zoomed area of Figure 1, there are two measure points on the estimated contour assigned to one measure point on the original contour. Therefore, two distances are calculated, one between the measure point on the original contour and the first measure point on the estimated contour and another between the measure point on the original contour and the second measure point on the estimated contour. These two distances are averaged, resulting in the distance value d_i for the investigated measure point i on the original contour.

2.2 Temporal Coherency

The temporal coherency of an estimated 2D-shape sequence is evaluated by the temporal variation of the spatial accuracy criteria between succeeding frames. This means, if the normalized mean $\overline{d}$ and the normalized standard deviation σ_d of the distance values d_i between the original and the estimated contour are similar for succeeding frames, the temporal coherency is judged as good, otherwise it is judged as bad. Thus, by these two criteria it can be detected if the normalized mean or the normalized standard deviation of the distance values between the original and the estimated contour changes for succeeding frames. However, it is not detected if the measured distances keep the same value in succeeding frames, but the spatial position of the measured values changes. This case would lead to a visually bad temporal coherency. In order to detect it a third criterion for evaluation of the temporal coherency is used, which is proposed in [8] (see Figure 2):

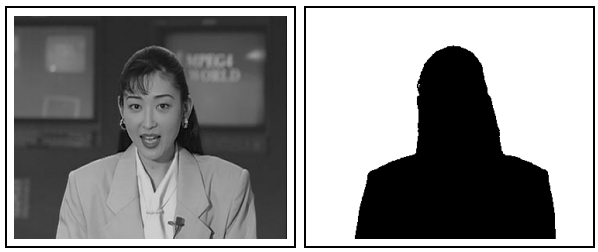
$$\Delta g_t = \left| \frac{1}{\emptyset_t} \left(\vec{g}_t^{ori} - \vec{g}_t^{est} \right) - \frac{1}{\emptyset_{t-1}} \left(\vec{g}_{t-1}^{ori} - \vec{g}_{t-1}^{est} \right) \right|$$

The vectors $\vec{g}_t^{ori}$ and $\vec{g}_t^{est}$ are the gravity centers of the evaluated object in the original and the estimated object mask at time instance t, respectively. Δg_t is the amount of variation from time instance t-1 to t of the by the maximal object diameter $\emptyset$ normalized difference between the gravity centers in the original and estimated object mask. If the position of estimation errors does not change between two frames the value of Δg_t is small. For this third criterion it is assumed that changes of the position of estimation errors are not symmetrically to the gravity center.

3 EXPERIMENTAL RESULTS

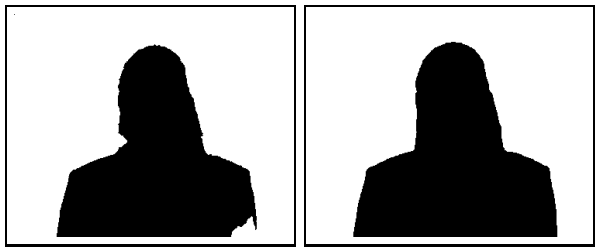
The proposed evaluation method has been applied to shape estimation results for several test sequences. Thereby, a good correspondence with the visual impression of the results was established.

With the results in Figure 3 it is demonstrated that the stated first problem and therefore also the second problem of the approaches from the literature are solved by the presented evaluation method. Figure 3a shows the original frame of the MPEG-4 test sequence *Akiyo*.



(a) Original frame

(b) Original object mask



(c) First example of an estimated object mask

(d) Second example of an estimated object mask

Figure 3: Exemplary 2D-shape estimation results for frame 30 of the MPEG-4 test sequence *Akiyo*

The corresponding original 2D-shape represented by an object mask is shown in Figure 3b. Figure 3c and Figure 3d present two exemplary shape estimation results. In the first one, which corresponds to case 2 in the introduction, a part of the left arm and a part of the hair of the person are missing. Thus, there are large estimation errors mainly at two positions of the object contour. The second one is the blowed original object mask, which therefore has a lot of small estimation errors around the object contour. This corresponds to case 1 in the introduction. Although both shapes look very different, they give similar values for the spatial accuracy, if evaluated by an approach from the literature, e.g. [5]. Using the evaluation method proposed in this paper, the two criteria for evaluating the spatial accuracy have the following values written in percent:

Estimation result	$\overline{d}$[%]	σ_d[%]
Object mask in Figure 3c	0.824	1.485
Object mask in Figure 3d	0.856	0.437

The normalized mean $\overline{d}$ of the estimation errors of both 2D-shapes is nearly equal. But, their normalized standard deviation σ_d is quite different. Therefore, the spatial accuracy of both results is judged different if using the proposed evaluation method. Furthermore, this improvement of the evaluation of the spatial accuracy has an impact on the evaluation of the temporal co-

herency, which means that the evaluation of the temporal coherency is improved, too.

In order to get an impression of the evaluation of a complete sequence of estimated 2D object shapes, in Figure 4 results for all criteria of the proposed evaluation method are shown for estimation results of the MPEG-4 test sequence *Akiyo* generated by the COST 211 Analysis Model 5.0 [3][1]: For the first eight frames of the sequence the two spatial criteria (Figures 4a–4b) and also the three temporal criteria (Figures 4c–4e) have quite large values, because it needs some frames, until the estimated 2D-shape covers the complete silhouette of Akiyo, and of course the estimated 2D-shape changes rapidly between these frames. For all following frames the 2D-shape of Akiyo is well estimated, which results in low values for the spatial criteria and also for the temporal criteria. At frames 51 and 67 the estimated 2D-shape has small estimation errors in the head area, which explains the small peaks in Figure 4.

4 CONCLUSIONS

In the literature, first approaches for objective evaluation of 2D-shape estimation results for moving objects in a video sequence are proposed. By these approaches the spatial accuracy and the temporal coherency of the estimated 2D-shapes is evaluated using the correct, original 2D-shapes as reference, which must be known. Thereby it is not distinguished, if an estimated 2D-shape has a lot of small estimation errors (case 1) or if it has only a few large estimation errors (case 2).

In this paper an evaluation method is proposed, which determines the mean and the standard deviation of the distances between an estimated 2D-shape and the corresponding original one measured in several contour points. For representing the spatial accuracy the mean and the standard deviation are normalized by the maximal diameter of the original 2D-shape.

It is shown that the normalized mean of the measured deviations between the original and an corresponding estimated 2D-shape is a useful criterion to evaluate the spatial accuracy. Furthermore, both cases, i.e. case 1 and case 2, are distinguished by the normalized standard deviation, which solves the above problem.

For evaluating the temporal coherency, the variation of the normalized mean and of the normalized standard deviation between succeeding frames is investigated. As by these two criteria a change of the spatial position of estimation errors is not considered, a third criterion is used, which evaluates the temporal variation of the gravity centers of the original and the estimated 2D-shape. This only works if the changes are not symmetrically with respect to the gravity center.

The approach has been tested with 2D-shape estimation results for several test sequences. Thereby, a good correspondence with the visual impression of the results was established. Of course, it is possible to combine this evaluation method with the ideas from [5][8] by distinguishing positive and negative measured distances that are weighted differently with given functions. Then, the proposed evaluation method can be adapted to the requirements of a specific application.

References

[1] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, T. Sikora, "Image Sequence Analysis for Emerging Interactive Multimedia Services - The European COST 211 Framework", in *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 7, November 1998, pp. 802-813.

[2] P. Correia, F. Pereira, "Objective Evaluation of Relative Segmentation Quality", in *Proc. International Conference on Image Processing 2000*, Vancouver, Canada, September 2000.

[3] M. Gabbouj, G. Morrison, F. Alaya-Cheikh, R. Mech, "Redundancy Reduction Techniques and Content Analysis for Multimedia Services - the European COST 211quat Action", in *Proc. Workshop on Image Analysis for Multimedia Interactive Services 1999*, Berlin, Germany, May/June 1999.

[4] B. Marcotegui, P. Correia, F. Marques, R. Mech, R. Rosa, M. Wollborn, F. Zanoguera, "A Video Object Generation Tool Allowing Friendly User Interaction", in *Proc. International Conference on Image Processing 2000*, Kobe, Japan, October 1999.

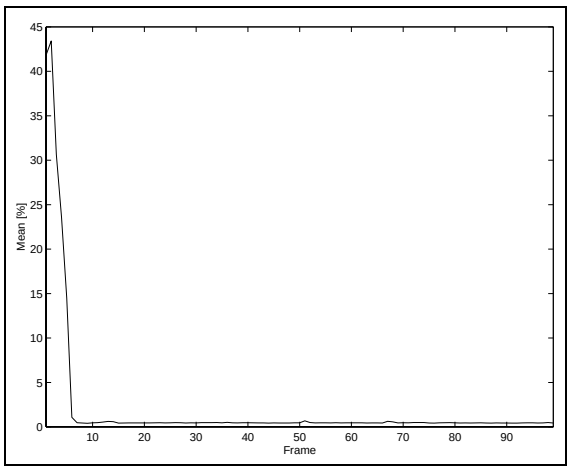
[5] X. Marichal, P. Villegas, "Objective Evaluation of Segmentation Masks in Video Sequences", in *Proc. of European Signal Processing Conference 2000*, Tampere, Finland, September 2000.

[6] MPEG-4: Doc. ISO/IEC JTC1/SC29/WG11 N2502, "Information Technology - Generic Coding of Audiovisual Objects, Part 2: Visual, Final Draft of International Standard", October 1998.

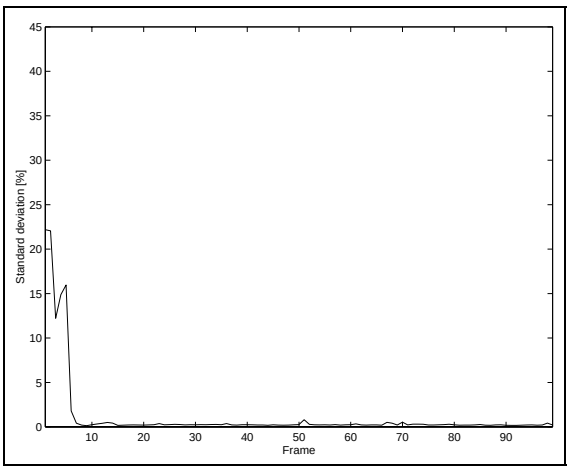
[7] Internal report of the University of Hannover: J. Radmer, "Fehleranalyse der Ergebnisse einer 2D-Formschätzung bewegter Objekte", Hannover, Germany, February 2001.

[8] P. Villegas, X. Marichal, A. Salcedo, "Objective Evaluation of Segmentation Masks in Video Sequences", in *Proc. Workshop on Image Analysis for Multimedia Interactive Services 1999*, Berlin, Germany, May/June 1999.

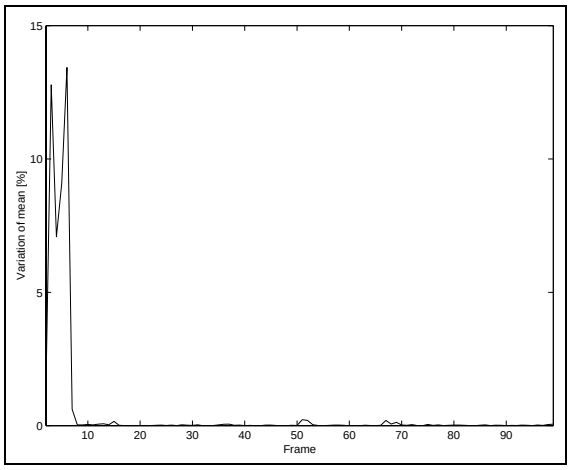
[9] M. Wollborn, R. Mech, "Procedure for Objective Evaluation of VOP Generation Algorithms", Doc. ISO/IEC JTC1/SC29/WG11 MPEG97/2704, Fribourg, Switzerland, October 1997.



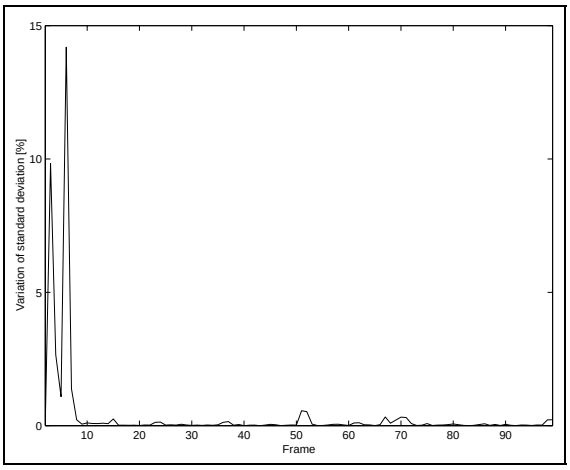
(a) Normalized mean of distances



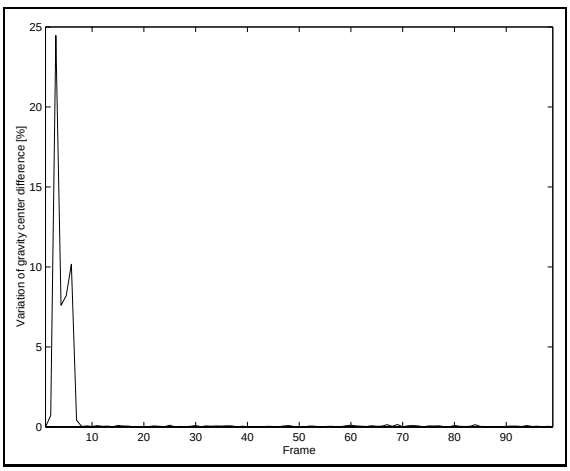
(b) Normalized standard deviation of distances



(c) Variation of normalized mean of distances



(d) Variation of normalized standard deviation of distances



(e) Variation of normalized gravity center difference

Figure 4: Evaluation of 2D-shape estimation results for the MPEG-4 test sequence *Akiyo* (30 Hz) generated by the COST 211 Analysis Model 5.0 using the proposed evaluation method.

STANDALONE OBJECTIVE EVALUATION OF SEGMENTATION QUALITY

Paulo Correia, Fernando Pereira

Instituto Superior Técnico - Instituto de Telecomunicações
Av. Rovisco Pais, 1049-001 Lisboa, PORTUGAL
E-mail: Paulo.Correia@lx.it.pt, Fernando.Pereira@lx.it.pt

ABSTRACT

The identification of objects in video sequences, i.e. video segmentation, plays a major role in emerging multimedia interactive services, such as those enabled by the ISO MPEG-4 and MPEG-7 standards. In this context, assessing the adequacy of the identified objects to the application targets, i.e. the evaluation of segmentation quality, assumes a crucial importance.

Video segmentation technology has received considerable attention in the literature, and several algorithms have been proposed to address various types of applications. However, the segmentation quality performance evaluation of those algorithms is often *adhoc*, and a well-established solution is not available. In fact, the field of objective segmentation quality evaluation is still maturing, and recently some efforts have been made, mainly following the MPEG object-based coding and description developments.

This paper discusses the problem of objective segmentation quality evaluation in its most difficult scenario: standalone evaluation, i.e. when a reference segmentation is not available for comparative evaluation. In particular, **objective metrics** are proposed for the evaluation of **standalone segmentation quality** for both individual objects and the overall segmentation partition.

1. INTRODUCTION

With the recent publication of the MPEG-4 standard [3], allowing to independently encode audiovisual objects, and the development of the MPEG-7 standard [7], allowing the content-based description of audiovisual material, the MPEG committee has given a significant contribution for the development of a new generation of interactive multimedia services. Innovative types of interaction are often based on the understanding of a video scene as composed by a set of video objects, to which it is possible to associate specific information as well as interactive "hooks" to deploy the desired application behaviour.

To enable such type of interactive services, an understanding of the scene semantics is required, notably in terms of the relevant objects that are present. It is in this context that video segmentation plays a determinant role. Segmentation may be automatically obtained at the video production stage, e.g. when using chroma keying techniques, or it may have to be directly obtained from the images captured by a camera through the usage of appropriate segmentation algorithms.

The evaluation of the adequacy of a segmentation algorithm, and its parameters' configuration, for a given application can be crucial to guarantee that the application interactive requirements can be fulfilled.

The current practice for segmentation quality evaluation mainly consists in the subjective *adhoc* assessment, by a representative group of human viewers. This is a time-consuming and expensive process, whose subjectivity can be minimised by following strict evaluation conditions, with the video quality evaluation recommendations developed by ITU providing valuable guidelines [4, 5].

Alternatively, objective segmentation quality evaluation methodologies can be used, even if the amount of attention devoted to this issue is not comparable to the investment on the segmentation algorithms themselves. Some proposals for objective evaluation have been made since the 1970's, mainly for assessing the performance of edge detectors [11]. More recently, the emergence of the MPEG-4 and MPEG-7 standards, has given a new impulse not only to the segmentation technology, but also to the segmentation quality evaluation methodologies – see for instance [8, 10]. However, the available metrics for segmentation quality evaluation typically perform well only for very constrained applications scenarios.

This paper discusses the objective evaluation of segmentation quality, in particular when no *ground truth* segmentation is available to use as a reference for comparison: **standalone evaluation**.

The various types of standalone segmentation quality evaluation are discussed in Section 2. Metrics for individual object and overall segmentation quality evaluation are proposed in Sections 3 and 4, respectively. Results are presented in Section 5 and conclusions in Section 6.

2. STANDALONE SEGMENTATION EVALUATION

Standalone segmentation quality evaluation is performed when no reference segmentation is available. Therefore, the *a priori* information that may be available about the expected segmentation results has a decisive impact on the type of evaluation to be performed, so that meaningful results can be

achieved. In particular, standalone evaluation of segmentation quality is not expected to provide as reliable results as the evaluation relative to a reference segmentation. A discussion on the relative evaluation of segmentation quality evaluation can be found in [2].

When performing **segmentation quality evaluation**, two types of measurements can be targeted:

- **Individual object evaluation** – When one of the objects identified by the segmentation algorithm is independently evaluated in terms of its segmentation quality.
- **Overall evaluation** – When the complete set of objects identified by the segmentation algorithm are globally evaluated as the components of the video sequence partition.

Objective segmentation quality evaluation uses automatic tools and thus produces objective evaluation measures. The automatic tools operate on segmentation results obtained for a selected set of sequences and, in the case of individual object evaluation, the object whose segmentation quality is to be assessed has first to be selected.

Overall segmentation quality evaluation requires the estimation of individual object quality, and the weighting of those values according to each object's relevancy in the scene, since segmentation errors in the more important objects are more noticeable to a human viewer. Additionally, the correct detection of the target objects should be checked.

Individual object segmentation quality evaluation is valuable when objects are independently manipulated, e.g. for reusing in different contexts. On the other hand, the overall segmentation quality evaluation may determine whether the segmentation algorithm is adequate for the application addressed.

Both the individual object and the overall segmentation quality measures can be computed for each time instant, requiring that some temporal processing of the instantaneous results is after done to reflect the segmentation quality over the complete sequence or shot. For instance, a temporal mean or median may be computed.

Building on the existing knowledge on segmentation quality evaluation and also on some relevant aspects from the video quality evaluation field, a set of relevant features to be evaluated for performing objective evaluation of standalone segmentation quality, and appropriate objective quality metrics for both individual objects and the overall segmentation partition are proposed in the following.

With standalone segmentation quality evaluation, significant assessment results are only expected for well-constrained applications, and these results mainly provide qualitative information for the ranking of segmentation partitions and algorithms.

3. INDIVIDUAL OBJECT EVALUATION

Metrics for individual object standalone segmentation quality evaluation can be established based on the expected homogeneity of each object's features (**intra-object features**), as well as on the observed differences of some key features against those of the neighbours (**inter-object features**).

Intra-object homogeneity can be evaluated by means of spatial and temporal object features, as discussed below.

The **spatial features** considered for individual object evaluation, and corresponding metrics, are:

- **Shape regularity** – Regularity of shapes can be evaluated by geometrical features such as the compactness (*compact*), or a combination of circularity and elongation (*circ_elong*) of the objects:

$$compact(E) = \max\left(\frac{perimeter^2(E)}{75 \cdot area(E)},\quad 1\right)$$

$$circ\_elong(E) = \max\left(circ(E),\quad \max\left(\frac{elong(E)}{5},\quad 1\right)\right)$$

  With circularity and elongation defined by:

$$circ(E) = \frac{4 \cdot \pi \cdot area(E)}{perimeter^2(E)}, \qquad elong(E) = \frac{area(E)}{(2 \cdot thickness(E))^2}$$

  Here *thickness(E)* is the number of morphological erosion steps that can be applied to the object until it disappears. The normalizing constants were empirically determined after an exhaustive set of tests.

- **Spatial uniformity** – Spatial uniformity can be evaluated by features such as spatial perceptual information (*SI*) [5], and texture variance (*text_var*) – see for instance [6].

The **temporal features**, and corresponding metrics, considered are:

- **Temporal stability** – A smooth temporal evolution of object features can be tested for checking temporal stability. These features include: size, position, temporal perceptual information [5], criticality [9], texture variance, circularity, elongation and compactness. The selected metrics for temporal stability evaluation are:

$$size_{diff} = |area(E_t) - area(E_{t-1})|$$

$$elong_{diff} = |elong(E_t) - elong(E_{t-1})|$$

$$crit_{diff} = |crit(E_t) - crit(E_{t-1})|$$

  With *crit(E)* being the criticality value as defined in [9].

- **Motion uniformity** – The uniformity of motion can be evaluated by features such as the variance of the object's motion vector values (*mot_var*), or by criticality (*crit*).

The above spatial and temporal features are not expected to be homogeneous for every segmented object; the applicability and importance of the corresponding metrics is conditioned by the type of application addressed.

Inter-object features give an indication if the objects were correctly identified as separate entities. These features can be computed either locally along the object boundaries, or for the complete object area. Again these features may be applicable only in some circumstances, such as when a significant contrast, or some feature value difference, between neighbouring objects is expected.

- **Local contrast to neighbours** – A local contrast metric can be used for evaluating if a significant contrast between the inside and outside of an object, along the object border, exists:

$$contrast = \frac{1}{4 \cdot 255 \cdot N_b} \cdot \sum_{i,j} \left(2 \cdot \max\left(DY_{ij}\right) + \max\left(DU_{ij}\right) + \max\left(DV_{ij}\right)\right)$$

  Where N_b is the number of border pixels for the object and DY_{ij}, DU_{ij}, and DV_{ij} are the differences between an object's border pixel Y, U and V components, respectively, and its neighbours.

- **Differences between neighbouring objects** – Several features, for which objects are expected to differ from their neighbours, can be tested. Examples are the shape regularity, spatial uniformity, temporal stability, and motion uniformity values, whenever each of them is relevant taking the application characteristics into account. In particular a metric for the motion uniformity feature is considered of interest:

$$mot\_unif_{neigh\_diff} = \frac{1}{N} \cdot \sum_{j \in NS_i} \left| mot\_unif_j - mot\_unif_i \right|$$

  Where i is the object under analysis, N and NS_i are, respectively, the number and the set of neighbours of object i, and the motion uniformity for each object is computed as:

$$mot\_unif_i = mot\_var + crit$$

Each of the elementary metrics considered for individual object segmentation quality evaluation is normalized to produce results in the interval [0, 1], with the highest values associated to the best segmentation quality results.

Since the usefulness of the various standalone evaluation elementary metrics has a strong dependency on the characteristics of the type of content/application considered, a single general-purpose composite metric cannot be established. Instead, the approach taken here is to select two major classes of content differing in terms of their spatial and temporal characteristics, and propose different composite metrics for each of them.

The two classes of content selected are:

- **Content class I: stable content** – Relevant for applications which content is temporally stable and have reasonably regular shapes. Additionally, the contrast between objects is expected to be strong.
- **Content class II: moving content** – Relevant for applications which content motion is rather important. Consequently, temporal stability is less relevant, motion uniformity is more significant and neighbouring objects may be spatially less contrasted, while their motion differences are more noteworthy. Regular shapes are still expected, even if assuming a lower importance.

3.1. Individual Object Metric for Stable Content

For content class I, stable content, a composite metric is proposed that excludes the elementary metrics related to spatial uniformity, as arbitrary spatial patterns may be found in the expected objects, and to motion uniformity, as motion is not very relevant in this case. Thus, the classes of elementary metrics considered for standalone individual evaluation of stable content are:

- *Shape regularity* –Two elementary metrics, compactness (*compact*) and a combination of circularity and elongation (*circ_elong*), are considered for evaluation of the shape regularity class.
- *Temporal stability* – Elementary metrics for the stability of size ($size_{diff}$), elongation ($elong_{diff}$) and criticality ($crit_{diff}$) are used to evaluate this class of metrics.
- *Local contrast to neighbours* – A local contrast metric (*contrast*) is considered for the evaluation of the contrast between neighbouring objects.

The proposed composite metric for standalone evaluation of segmentation quality, for content class I, (*Seg_qual_std_stable*) is the temporal average of the corresponding instantaneous values of ($Inst\_seg\_qual\_std\_stable_t$), given by:

$$Inst\_seg\_qual\_std\_stable_t = intra + inter$$

With:

$$intra = 0.30 \cdot \left(0.5 \cdot circ\_elong + 0.5 \cdot compact\right) + 0.33 \cdot \left(0.33 \cdot size_{diff} + 0.33 \cdot elong_{diff} + 0.33 \cdot crit_{diff}\right)$$

and:

$$inter = 0.37 \cdot contrast$$

3.2. Individual Object Metric for Moving Content

For content class II, the composite metric again includes only the relevant classes of elementary metrics. In this case, the content is not expected to be temporally stable, but the objects should have uniform motion, and the neighbouring objects motion differences should be pronounced. The classes of metrics considered for the standalone segmentation quality evaluation of this type of content are:

- *Shape regularity* – The same elementary metrics, *compact* and *circ_elong*, are again used, even if, due to motion, the shape regularity assumption may sometimes not be completely verified.
- *Motion uniformity* – The criticality metric (*crit*) is used to evaluate whether objects exhibit a reasonably uniform motion.
- *Local contrast to neighbours* – Even if contrast is not so important in terms of segmentation quality evaluation as for stable content, the local contrast (*contrast*) metric is yet considered useful.
- *Difference between neighbouring objects* – Since neighbouring objects are expected to exhibit different motion characteristics, the motion uniformity difference metric (mot_unif_{diff}) is used.

The proposed composite metric for content class II, (*Seg_qual_std_moving*) is the temporal average of the corresponding instantaneous values ($Inst\_seg\_qual\_std\_moving_t$), given by:

$$Inst\_seg\_qual\_std\_moving_t = intra + inter$$

With:

$$intra = 0.32 \cdot (0.5 \cdot circ\_elong + 0.5 \cdot compact) + 0.31 \cdot crit$$

$$inter = 0.11 \cdot contrast + 0.26 \cdot mot\_unif_{neigh\_diff}$$

4. OVERALL SEGMENTATION EVALUATION

The objective overall segmentation quality evaluation combines the individual evaluation of each object's segmentation quality, with the corresponding relevance value and a factor reflecting the similarity between the target and the estimated objects.

Individual object evaluation has been specified in the previous Section. The relevance of objects is evaluated using a metric called *Relevance_context*, which has been proposed in [1]. This metric computes a relevance value reflecting how much the human viewer attention is attracted by a given object, and produces results in the [0,1] range, with the restriction that the relevancies of all objects composing a partition at a given time instant sum to one. Value one corresponds to the highest possible relevance.

The assessment of the similarity of objects for standalone segmentation quality evaluation, and the computation of the overall segmentation quality metric are described below.

4.1. Similarity of Objects Evaluation

The similarity of objects is evaluated by computing a metric called *Sim_obj_factor*, which is a multiplicative factor to be included in the overall segmentation quality evaluation metric.

In standalone evaluation, this similarity is mainly reduced to an evaluation about the correctness of the number of objects detected, if this information is available. The corresponding metric (*num_obj_comparison*) is defined by:

$$num\_obj\_comparison = \frac{\min(num\_est\_obj, num\_target\_obj)}{\max(num\_est\_obj, num\_target\_obj)}$$

Where *num_est_obj* and *num_target_obj* are the numbers of estimated and target objects, respectively.

This metric provides a limited amount of information, in particular not distinguishing between too many or too few detected objects. To make the *Sim_obj_factor* metric more informed, it is possible to consider also a measure of the number of objects stability (*num_obj_stability*), applicable whenever the evolution of the segmentation partition is assumed to be smooth:

$$num\_obj\_stability = \frac{\min(num\_obj_{t-1}, num\_obj_t)}{\max(num\_obj_{t-1}, num\_obj_t)}$$

Where num_obj_t is the number of estimated objects in time instant *t*.

The proposed *Sim_obj_factor* metric for standalone segmentation quality evaluation is thus obtained by complementing the *num_obj_comparison* factor with the *num_obj_stability* factor:

$$Sim\_obj\_factor = num\_obj\_comparison \cdot num\_obj\_stability$$

Whenever one of the two factors above cannot be computed, or is not applicable, only the other is considered. Additionally, since the two factors vary as time evolves, a *Sim_obj_factor* representative of the complete sequence is obtained by a temporal average of the instantaneous values.

4.2. Overall Segmentation Quality Metric

The computation of the overall segmentation quality metric, both for standalone and relative evaluation, combines the appropriate measures of individual object quality, their relevance and the similarity of objects factor. The proposed metric is computed by:

$$Seg\_qual = Sim\_obj\_factor \cdot \left(\sum_i (Seg\_qual\_ind(E_i) \cdot Relevance\_context(E_i)) \right)$$

Where $Seg\_qual\_ind(E_i)$ is the individual segmentation quality for object *i*, $Relevance\_context(E_i)$ is the corresponding relevance, and *Sim_obj_factor* is the factor evaluating the correspondence between the detected and target objects. The sum is performed for all the estimated objects.

Instead of including the temporal dimension influence separately in each factor, as presented so far, a weighting of the instantaneous objects' quality by their instantaneous relevance values, and then by the instantaneous similarity of objects factor, to reflect the variations in quality, relevance or similarity values that may occur along time, is used.

With this overall segmentation quality evaluation metric, the higher the individual object quality is for the most relevant objects, the better is the resulting quality evaluation. Therefore, the most relevant objects, which are the most visible to the human observers, have a larger impact on the overall segmentation quality evaluation. Furthermore, if a correct match between target and estimated objects is not achieved, then a penalizing factor is correspondingly included.

5. RESULTS

Results obtained with the metrics proposed in the previous Sections for standalone segmentation quality evaluation are discussed below, after presenting a set of test sequences and corresponding segmentation partitions.

5.1. Test Sequences and Segmentation Partitions

A series of tests of the proposed segmentation quality evaluation metrics has been performed, using several test sequences, mainly from the MPEG-4 test set, showing different spatial complexity and temporal activity characteristics. For each sequence, several segmentation partitions with different segmentation qualities were considered.

Two subsets of the test sequences, each with 30 representative images of the desired object behaviour and characteristics, are used to illustrate the obtained results. These subsequences are:

- ***Akiyo***, images 0 to 29 – This is a sequence with low temporal activity and not very complex texture. It contains two objects of interest: the woman, and the background.
- ***Stefan***, images 30 to 59 – This is a sequence with high temporal activity and relatively complex texture. It contains two objects of interest: the tennis player, and the background.

Sample original images and segmentation partitions are shown in Figure 1 and in Figure 2, respectively for the *Akiyo* and *Stefan* sequences. The segmentation partitions labelled as *reference* are those made available by the MPEG group, and the other partitions were created with different segmentation quality levels, ranging from a close match with the reference to more objectionable segmentations.

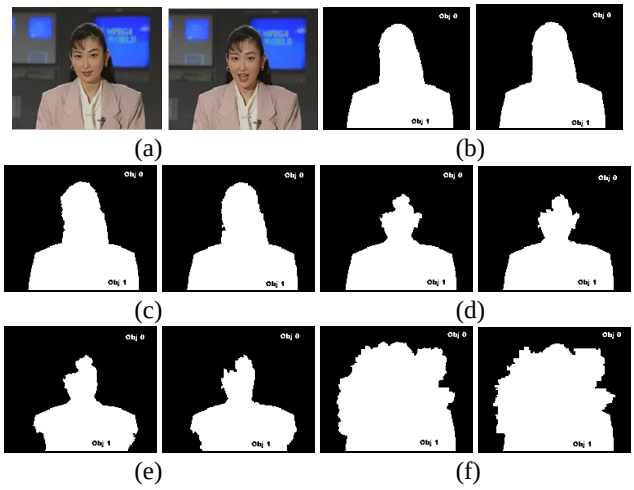


Figure 1 – Sample original images (a) and segmentation partitions: *reference* (b), *seg1* (c), *seg2* (d), *seg3* (e), and *seg4* (f) for images number 0 and 29 of the *Akiyo* sequence

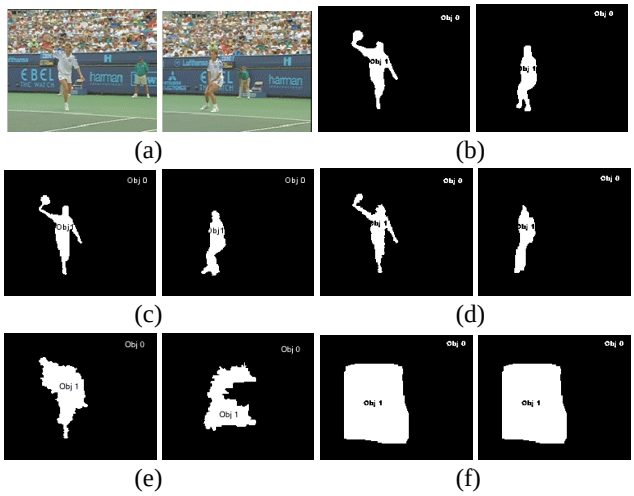


Figure 2 – Sample original images (a) and segmentation partitions: *reference* (b), *seg1* (c), *seg2* (d), *seg3* (e), and *seg4* (f) for images number 30 and 59 of the *Stefan* sequence

5.2. Standalone Segmentation Quality Evaluation Results

Standalone segmentation quality evaluation metrics are applicable only in certain circumstances, and thus the two metrics proposed have been tested with the appropriate contents. Results for these metrics, considering both the **individual object** and the **overall** evaluation cases are included below.

A set of preliminary experiments showed that similar segmentation quality evaluation results are produced independently of the input format, and thus the QCIF resolution was used to limit the algorithm execution time.

For each test sequence, the results include a graph, representing the temporal evolution of the overall segmentation quality, and a table, containing the temporal average of the instantaneous results computed both for individual object and for overall segmentation quality evaluation.

Content class I corresponds to video sequences which have relatively simple shapes, and present a limited amount of motion. To evaluate this type of content, the *Akiyo* test sequence and the corresponding segmentation partitions were used.

The results, included in Figure 3, show that for the *woman* object there are three segmentation quality groups: best quality is achieved by the reference, segmentation *1*, and segmentation *2*, then segmentation *3* achieves intermediate values, and finally segmentation *4* gets the worst results. These results agree generally agree with those a human observer would produce. The reference segmentation does not get the best result since a part of the woman's hair is intensely illuminated, and when included as part of the woman it leads to a lower contrast to the background than when it is omitted, as it happens with segmentations *1* and *2*. Segmentation *4*, for which the *woman* object captures a significant part of the *background*, is clearly identified as the worst segmentation.

	Average Segment. Quality		
	Back.	Woman	Overall
Ref	0.76	0.77	0.77
Seg1	0.79	0.79	0.79
Seg2	0.79	0.80	0.80
Seg3	0.73	0.73	0.73
Seg4	0.65	0.56	0.60

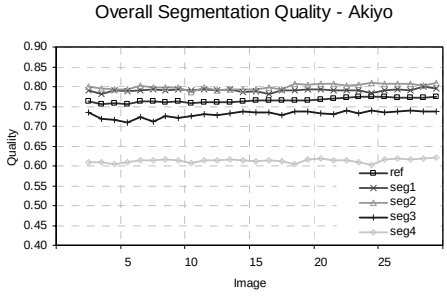


Figure 3 - Standalone overall and individual object quality evaluation results for the *Akiyo* sequence

Content class II corresponds to more complex video content than in the previous case. Object shapes may not be so simple, and motion may be more important. The *Stefan* sequence and the corresponding segmentation partitions were used to evaluate the proposals made in this paper with this type of content.

The results, included in Figure 4, show that segmentation *1* gets the best overall segmentation quality result followed by a group formed by the reference and segmentations *2* and *3*. Segmentation *4* gets the worst result. These results can be explained as follows. Segmentation *1* is more precise than the reference partition, as the reference is smoother and sometimes includes small parts of the *background* as belonging to the *player* object. The reference and segmentation *2* are correctly classified as the next quality group, but segmentation *3* should be ranked as having lower quality than these two. Finally, segmentation *4* is correctly ranked as the worst, since it is static and includes a large amount of *background* into the *player* object, but its quality value is higher than expected.

	Average Segment Quality		
	Back.	Player	Overall
Ref	0.33	0.43	0.38
Seg1	0.34	0.49	0.42
Seg2	0.32	0.43	0.38
Seg3	0.34	0.45	0.39
Seg4	0.32	0.37	0.34

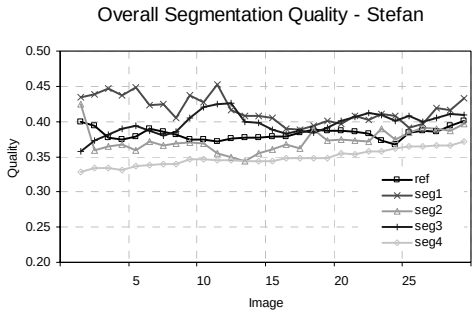


Figure 4 - Standalone overall and individual object quality evaluation results for the *Stefan* sequence

The above examples demonstrate that the standalone segmentation quality evaluation algorithm is capable of ranking the quality of the various segmentation partitions, but the results must be interpreted in a rather qualitative and relative way (e.g. for ranking purposes). Standalone evaluation results are not expected to be as reliable as those of relative evaluation, but they can still be useful for identifying several quality groups among the various tested segmentations/algorithms.

6. CONCLUSIONS

Segmentation quality evaluation is a key element whenever the identification of a set of objects in a video sequence is required since it allows the assessment of the performance of segmentation algorithms in view of a given application targets.

Since a satisfying solution for segmentation quality evaluation is not yet available, this paper discusses the problem, in particular when a reference segmentation playing the role of *ground truth* is not available – standalone evaluation.

Metrics for both individual object and for overall segmentation quality evaluation were proposed. As expected, standalone evaluation revealed itself sensitive to the type of application/content considered. The various classes of elementary metrics available are not universally applicable, but when carefully selected metrics are employed for given classes of content then useful segmentation quality evaluation results can be obtained. Nevertheless, these evaluation results have a more qualitative rather than quantitative value, mainly\ allowing relative comparisons of segmentation results.

7. REFERENCES

[1] P. Correia, F. Pereira; "Estimation of video object's relevance", *EUSIPCO'2000*, Finland, Sept. 5-8, 2000, pp. 925-928

[2] P. Correia, F. Pereira; "Objective Evaluation of relative segmentation quality", *ICIP'2000*, Canada, Sept.10-13, 2000, pp. 308-311

[3] ISO/IEC 14496, "Information technology - Coding of audio-visual objects", 1999

[4] ITU-R, "Methodology for the subjective assessment of the quality of television pictures", *Recommendation BT.500-7*, 1995

[5] ITU-T, "Recommendation P.910 - Subjective video quality assessment methods for multimedia applications", August 1996

[6] M. Levine, A. Nazif, "Dynamic measurement of computer generated image segmentations", IEEE Trans. Pattern Analysis Machine Intelligence, Vol. PAMI-7, No.2, March 1985, pp.155-164

[7] MPEG Requirements Group, "MPEG-7 overview", Doc. ISO/IEC JTC1/SC29/WG11 N4031, Singapore MPEG meeting, March 2001

[8] P. Villegas, X. Marichal, A. Salcedo, "Objective evaluation of segmentation masks in video sequences", *WIAMIS'99*, Germany, 31 May - 1 June 1999, pp. 85-88

[9] S. Wolf, A. Webster, "Subjective and objective measures of scene criticality", *ITU Meeting on Subjective and Objective Audiovisual Quality Assessment Methods*, Turin, October 1997

[10] M. Wollborn, R. Mech, "Refined procedure for objective evaluation of video object generation algorithms" *Doc. ISO/IEC JTC1/SC29/WG11 M3448*, March 1998

[11] Y. J. Zhang; "A survey on evaluation methods for image segmentation", *Pattern Rec., Vol. 29(8)*, 1996, pp. 1335-1346

Immersive Communication

Damien Douxchamps, David Ergo, Benoît Macq, Xavier Marichal,
Alok Nandi, Toshiyuki Umeda, Xavier Wielemans
alterface *
c/o Laboratoire de Télécommunications et Télédétection
Université catholique de Louvain
2 place du Levant
B-1348 Louvain-la-Neuve - Belgium
Contact e-mail: marichal@tele.ucl.ac.be

ABSTRACT

An apparatus for communication/entertainment mixing synthetic and natural images in real-time is designed and allows the "user" to be captured through vision-based sensors, like (web) cameras. The composed visual scenes are to be experienced in physical spaces and/or to be viewed through web browsers. The word "transfiction" has been coined to this interactive narrative system where users can interact with narrative machines (devices with computing power and containing databases of meaningful information).

1 Introduction

Contrary to many approaches to virtuality or mixed reality, the designed system does not need any dedicated hardware, nor for computation nor for tracking of real objects/persons. It runs on standard Pentium PCs and cameras are the only used sensors. This vision-based interface approach allows complete freedom to the user, not anymore tied to hardware devices such as helmets and gloves. Various research projects have already adopted such a user-centric approach towards mixed reality. It ranges from the only animation/command of purely virtual worlds, as in the KidsRoom [1], to more mixed worlds where users see a virtually reproduced part of themselves as in N.I.C.E. [2], and goes to the inclusion of the user image within the virtual space in order to fully exploit the potential of mixed reality. In ALIVE [3], "Artificial Life Interactive Video Environment", wireless full-body interaction between a human participant and a rich graphical world inhabited by autonomous agents is used.

The present system of "transfiction" [4] aims at extracting users out of their context when they enter the space of some camera. The image is analyzed, the visual representation of people is automatically extracted and then integrated within a pre-existing story in order to construct the mixed-reality scene, as depicted in figure 1. The users' attitudes and behaviors then influence the narrative, with the explicit intent of making the immersion a rich experience for all users.

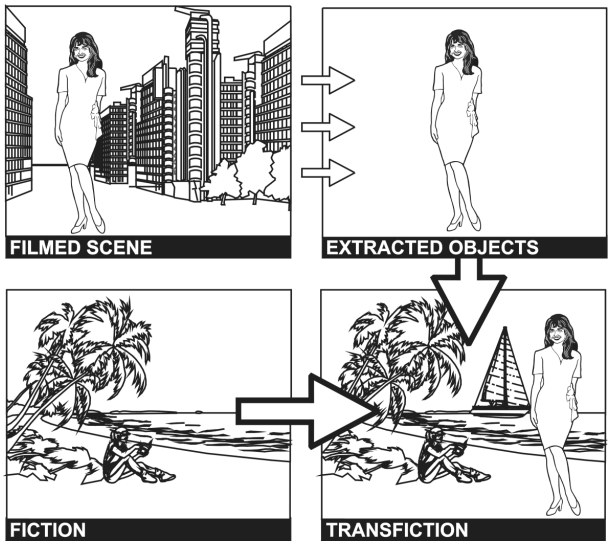


Figure 1: Underlying concept of transfiction

Since it makes use of both real and graphical images, transfiction needs to be positioned in the real-virtual continuum and in the context of the "mixed reality". According to Milgram and Colquhoun [5], mixed reality covers the whole continuum ranging from reality to virtuality. At the one end is the real environment, made of the real world and image capture of it. On the other end is the virtual environment, i.e. a world completely modeled in terms of shape, location, texture, motion... Mixed reality consists thus of any combination of these two worlds. According to the relative importance of real or virtual (modeled) elements, one has to deal with augmented reality or augmented virtuality as depicted on figure 2.

The reminder of the present paper is organized as follows. Sections 2 and 3 summarizes some key features of the transfiction [4] system: section 2 elaborates on some of its key concepts while section 3 presents the main as-

* Part of the research presented in the present paper is achieved in the framework of the *art.live* project (http://www.tele.ucl.ac.be/projects/art.live). This project is part of the fifth framework programme of the European Commission: Information Society Technologies (IST, project 10942).

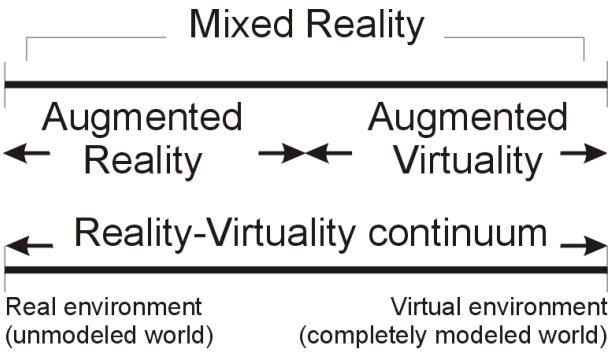


Figure 2: Mixed-Reality framework

pects of the underlying architecture. Finally, section 4 presents some results that are used to explain and illustrate the way interaction makes the scenario progress.

2 Concepts

Considering a human-centric approach, the various "users" depicted on figure 3 are involved within the "transfiction" system.

They are ranked here by their degree of influence on the overall system:

1. The *Author*, who designs the whole narrative system, i.e. the scenario, the related story and its elements (musical analogy to composer or theatre/film analogy to the scriptwriter);

2. The *Director*, who can modify (via the authoring tool) some aspects that the author prepared to be modifiable (musical analogy to performer or theatre/film analogy to the director achieving the mise-en-scene);

3. The *Consumer-Interactor*, who is captured by some live camera, and can directly interact with the system via its gesture. The Interactor is aware of his/her role in the narrative thanks to some (large) screen where s/he sees himself/herself within the mixed-reality environment;

4. The *Consumer-Player*, who interacts with the system through a mouse on a Web browser (clicking on some MPEG-4 hypervideo);

5. The *Actor*, who is any person in front of some live camera. The Actor is not aware of his/her status within the system;

6. The *Spectator*, who is any person looking at the images without interacting or being captured by the cameras.

It is important to stress that the difference between an interactor and an actor only resides in the degree of awareness of the user itself. Basically, both these users are in front of some cameras and have some influence on the system because of their attitude. Since the interactor is really made aware of the impact of his own behavior thanks to the big screen, it is expected that he will not remain a passive actor anymore but will adopt specific gesture and attitude in order to interact with the system.

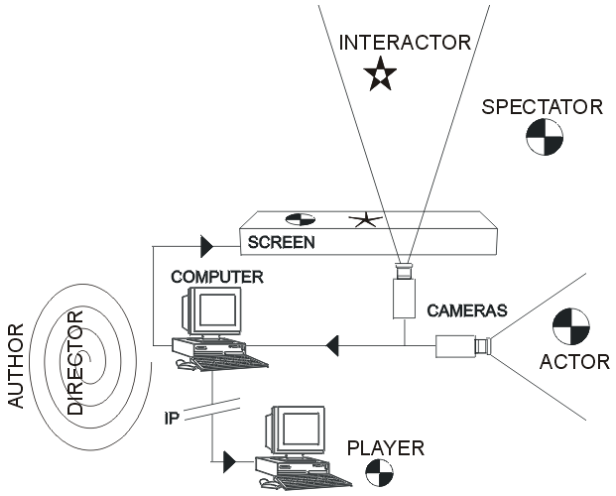


Figure 3: Repartition of users in the system

In figure 3, a systemic approach has been adopted: users are bound to a narrative apparatus which consists of cameras for image capture, computers for image composition, signal processing and scene composition, and screens for image rendering. For the sake of clarity, it is important to provide a praxis-based typology of the different spaces in which the bodies, objects and events are taking place. With this respect, it is interesting to quote Deleuze [6] who already elaborated on such considerations about different types of spaces: "We opposed the virtual and the real: although it could not have been more precise before now, this terminology must be corrected. The virtual is opposed not to the real but to the actual. The virtual is fully real in so far as it is virtual. Exactly what Proust said of states of resonance must be said of the virtual: 'Real without being actual; ideal without being abstract'; and symbolic without being fictional." Relying on such a point of view, the following definitions are used:

- The *Actual Space* is the space in front of the camera. It is the space in which any person becomes an interactor.

- The *Real Space* is the space in which the user is living, be it consumer, spectator, author....

- The *Virtual Space* is the space which is rendered on the screens. It is composed of real-time images of the interactors or other real elements as well

as bodies and objects generated from a computer database.

- The *Diegetic Space* (which is more specific for narrative films) refers to the world of a film story. The diegesis includes events that are presumed to have occurred as well as actions and spaces not shown onscreen. The concept of diegesis will take its plain dimension once we will be able to offer to the audience an extended narrative experience similar to viewing a film.

3 Technical Architecture

In order to provide users with such an experience, the technological challenge is to gather all needed subcomponents and issue a real-time implementation of the system. To compose all the visual objects (the "real" and computer-generated ones) within the final mixed-reality scene and to allow for interactivity, the MPEG-4 standard [7] can be used as the transmission layer.

In addition to the composition and transmission facilities, the following issues are addressed:

- Achieving a real time segmentation of moving objects captured by a web camera. As in [8], change detection is combined with automatic background adaptation in order to provide fast but robust object extraction.

- Associating these objects extracted from the real world with predefined synthetic objects in order to compose a coherent mixed-reality scene. Images, with a transparency layer, animations and short movies can be used. However, the *author* must pay extreme attention to the overall coherency of the combination of the different layers.

- Performing the real-time encoding of the arbitrarily shaped objects with the help of various coding techniques that appear within the MPEG-4 framework.

- Establishing a client-server architecture based on the Internet Protocol that allows for ubiquity (simultaneous re-use of the same camera images within various mixed-reality scenes) and composition of scenes from various local or remote sources.

- Automatically extracting (MPEG-7 [9] like) descriptors that are used to describe the behavior of visual objects. Such descriptors are presented with some more details in section 4 and are used to pilot the interactive scenario.

Solutions to these issues have been combined in order to implement the architecture depicted on figure 4.

Thanks to the Internet Protocol, the system is very flexible and allows any screen to access any resource it needs. A phenomenon of ubiquity is therefore provided

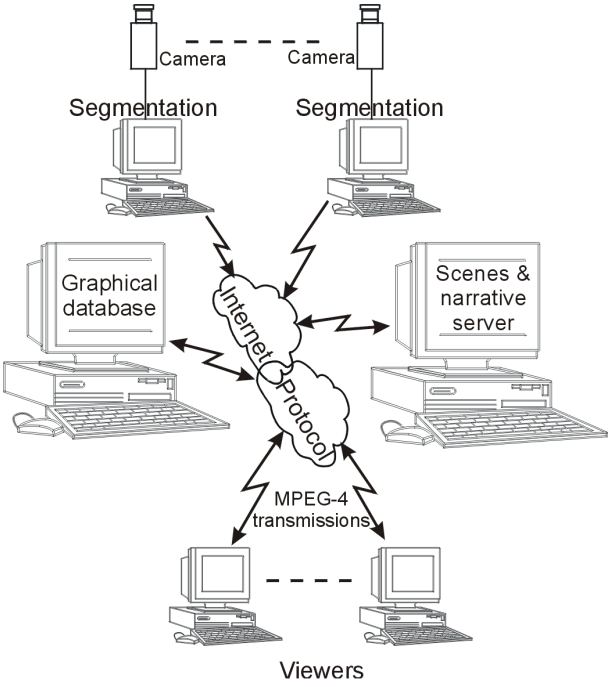


Figure 4: Technical architecture

since two or more screens may simultaneously access the same camera stream. Therefore, the system is very open and any device can be as much reproduced as needed.

4 Interactive Scenario

As already mentioned, descriptors are used for managing the application and offering users the possibility to interact with the scenario. Therefore, descriptors are attached to some of the graphical elements. For instance, on figure 5, one can see that three 'touchzones' are defined at three particular locations of the graphical elements.

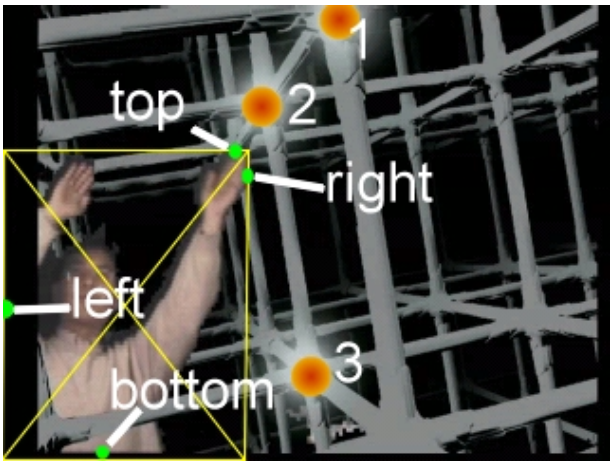


Figure 5: Presence of different descriptors in the current scene

These descriptors are compared in real-time with the ones generated by the interactor: in the present case, his/her image is described in terms of the bounding box (cf. figure 5) and contact points on the side of the box.

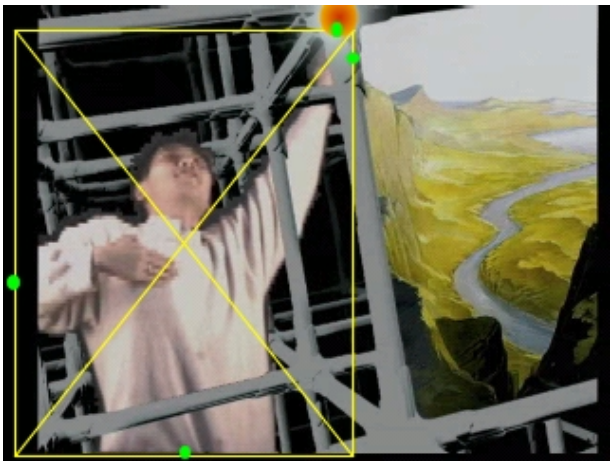


Figure 6: Activation of a scenario event thanks to descriptors

The location of one of these contact points (top - bottom - left - right) with some 'touchzones' causes an event in the scenario. On figure 6, the first 'touchzone' of figure 5 has been activated by the top contact point, provoking the appearance of another layer of graphics. 'Touchzone' number 2 was also to be activated by the top contact point while zone number 3 was targeted for the right contact point.

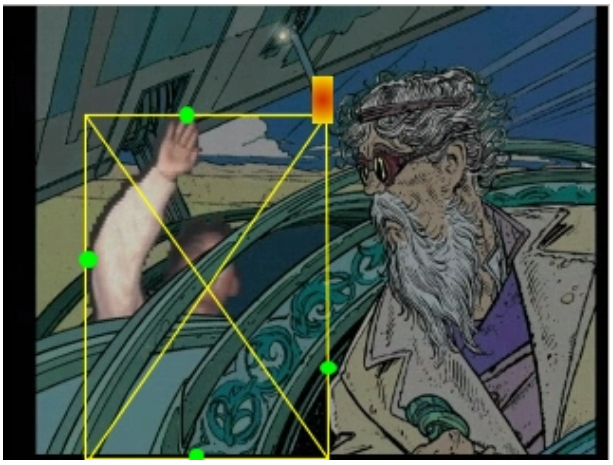


Figure 7: Although the bounding box is in contact with the 'touchzone', no event is activated

As shown on figures 7 and 8 the use of contact points instead of the bounding box itself allows increasing the robustness of the system with respect to noise and segmentation defaults while ensuring that interaction is really driven by the user. Indeed, it is very likely that his/her head and hands (interactive elements *per excellence*)are the elements in contact with the bounding box, therefore driving the application.

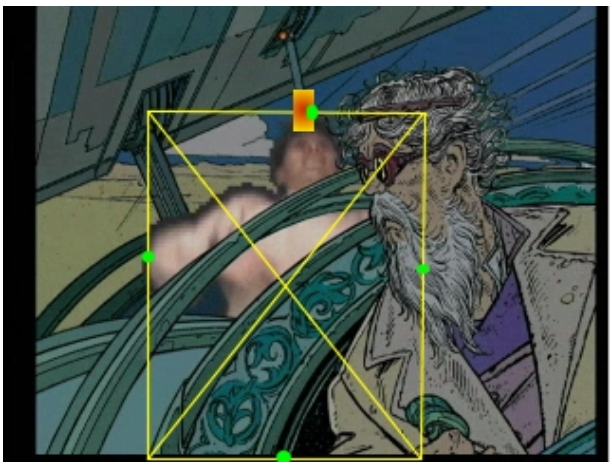


Figure 8: Activated event because of the presence of the top contact point in the 'touchzone'

Of course, intrinsic parameters of the bounding box, as well as other descriptors (like position, texture, motion) can be used to generate more events. For instance, on figure 9, the user (appearing behind the semi-transparent *drosera*, is offered to open or close the *drosera* itself according to the width of the bounding box. If he/she manages to open it completely, another event will occur, i.e. a change of scene and the pursuing of the interactive story.

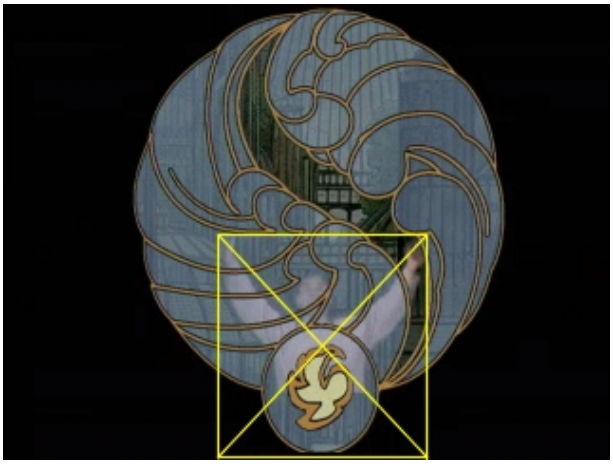


Figure 9: Interactive piloting of some animation (opening of the *drosera*)

References

[1] A. Bobick, S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schtte and A. Wilson, "The KidsRoom: A Perceptually-Based

Interactive and Immersive Story Environment", PRESENCE: Teleoperators and Virtual Environments, 8(4), August 1999, pp. 367-391.

[2] M. Roussos, A. Johnson, J. Leigh, C. Vasilakis, C. Barnes and T. Moher, "NICE: Combining Constructionism, Narrative, and Collaboration in a Virtual Learning Environment", Computer Graphics, 31(3), August 1997, pp. 62-63.

[3] P. Maes, T. Darrell, B. Blumberg and A. Pentland, "The ALIVE system: Wireless, full-body interaction with autonomous agents", Multimedia Systems, 5(2), March 1997, pp. 105-112.

[4] Alok Nandi, Xavier Marichal, "Transfiction", Virtual Reality International Conferences 2000, Laval, 18-21 May 2000.

[5] P. Milgram and H. Colquhoun, "A Taxonomy of Real and Virtual World Display Integration", in Mixed Reality, Merging Real and Virtual Worlds, Y. Ohta and H. Tamura editors, Ohmsha-Springer, 1999.

[6] G. Deleuze, "Difference and Repetition", London, Athlone, 1994.

[7] MPEG Video group, "MPEG-4 Video Verification Model, version 11.0", ISO/IEC JTC1/SC29/WG11 (N2178), Tokyo, March 1998.

[8] X. Marichal, "On-line Web Application using Image Segmentation", WIAMIS'99, Berlin, May 1999, pp. 141-144.

[9] MPEG group, "MPEG-7 description definition language document", version 2.0, ISO/IEC JTC1/SC29/WG11 (N2997), Melbourne, October 1999.

A 3-STEP ALGORITHM USING REGION-BASED ACTIVE CONTOURS FOR VIDEO OBJECTS DETECTION

S. Jehan-Besson[1], *M. Barlaud*[1], *G. Aubert*[2]
[1]Laboratoire I3S, CNRS UNSA, 2000 route des Lucioles
06903 Sophia Antipolis, France
[2]Laboratoire J.A. Dieudonné, CNRS, Parc Valrose
06108 Nice Cedex 2, France
e-mail: jehan@i3s.unice.fr

ABSTRACT

In this paper, we propose two contributions for video object segmentation.
First, we introduce a very full general framework for region-based active contours with a new Eulerian method to compute the evolution equation of the active contour. This framework can be easily adapted to various applications thanks to the introduction of functions named *"descriptors"* of the different regions.
Second, we propose a 3-step algorithm for detection of moving objects, with a static or a mobile camera, using region-based active contours. The active contour evolves with successively three sets of descriptors: a temporal one, and then two spatial ones. User interaction is reduced to the choice of a few parameters at the beginning of the process. The method has been evaluated on real sequences.

1 INTRODUCTION

Detection and localization of video objects become a crucial issue for the development of the new video coding standard MPEG-4 [1]. Indeed, in this new standard, functions such as manipulating, searching and interacting with meaningful video objects are required.

Generally speaking, two classes of approaches can be considered for segmentation: region-based approaches such as Markov Random Fields [2], and boundary-based approaches. Originally, active contours were boundary-based methods: snakes [3], or geodesic active contours [4] which are driven by the minimization of an energy towards the edges of an image. The information used is strictly along the boundary. In order to use active contours for the segmentation of moving objects, region-based information must be incorporated in the evolution equation of an active contour.

In this paper, we propose a generalized Eulerian framework for region-based active contours. This general framework is then applied to the detection of moving objects from video sequences acquired with either a moving or a static camera.

We first introduce a very full general framework for region based active contours. This framework can be easily adapted to various applications. Starting from a criterion including both region-based and boundary-based terms, we propose a new efficient Eulerian method to compute the evolution equation of an active contour. This proof ensures the fastest decrease of the active contour towards a minimum of the criterion. In the criterion, each region is described by a function named "descriptor" of the region. For a particular application, the user only has to choose well-adapted descriptors.

The second contribution of this paper is to propose a 3-step algorithm for detection of moving objects using the previous general framework. We propose to make the active contour evolve with sucessively three sets of descriptors. The first set is motion-based while the two others use spatial informations, namely edges and partition of the image in intensity homogeneous regions. The first stage enables to detect moving objects while the two others refine the result. In our method, the user does not have to select the object. User interaction is reduced to the choice of a few parameters at the beginning of the segmentation process. As far as sequences with mobile cameras are concerned, we propose to jointly perform camera motion compensation and segmentation. The camera motion model is directly included in the criterion to minimize. It is estimated from frame to frame using potential functions and the half quadratic theorem [5] for robust estimation.

The general framework for region-based active contours is detailed in section 2 while the 3-step algorithm for video objects detection is explained in section 3. Section 4 illustrates the potential of our approach by applying it to real sequences.

2 REGION-BASED ACTIVE CONTOURS

The main idea of this part lies in the development of a general framework for the segmentation of an image in two different regions using region-based active contours. We want to find the image partition that minimizes a criterion including both region-based and boundary-based terms. In this framework, each region is described using a function that we name "descriptor of the region". The introduction of such descriptors is interesting for two

reasons. First, for a given application like detection of moving objects, various descriptors can be easily tested inside the same theoretical framework or hierarchically combined as in section 3. Second, this framework can be applied to other applications [6].

Some authors [7, 8, 9, 10, 11] have proposed a way of adding region-based terms in the evolution equation of an active contour. These pioneer works are complementary and show the potential of region-based approaches. However, they are made for particular applications with particular descriptors. Moreover, all proofs leading to the evolution equation of the active contour are based on the derivation of the criterion using Euler-Lagrange equations [7, 10] and the dynamical scheme is introduced after the computation of the derivative. With such a method, the case of descriptors depending upon features globally attached to the region (time-dependent descriptors) cannot readily be taken into account.

In this paper, we introduce a general framework and we propose a new Eulerian method for the computation of the evolution equation of the contour. The major contribution of our work is to compute the evolution equation of the active contour without computing the Euler-Lagrange equations. We propose the powerful idea of introducing a dynamical scheme on the criterion itself, an a theorem of fluid dynamic to compute the derivative of the criterion. With such a method, the case of descriptors depending on the evolution of the curve, i.e. depending upon features globally attached to the region, can readily be taken into account, see [12] for details. The proof ensures the fastest decrease of the active contour towards a minimum of the criterion.

2.1 Introduction of a general criterion

Let I_n be the intensity of image number n in the sequence and Ω_n the image domain of frame number n. The image domain is considered to be made up of two parts: the foreground part, containing the objects to segment, noted $\Omega_{n,in}$, and the background part noted $\Omega_{n,out}$. The discontinuities set is a curve noted Γ_n that defines the boundary between the two domains.

We search for the two domains $\Omega_{n,in}$ and $\Omega_{n,out}$ which minimize the following criterion J_n:

$$J_n = \iint_{\Omega_{n,out}} k^{(n,out)} + \iint_{\Omega_{n,in}} k^{(n,in)} + \int_{\Gamma_n} k^{(n,b)} \qquad (1)$$

The first two terms are region-based while the third term is boundary-based. The functions $k^{(n,out)}(\cdot)$, $k^{(n,in)}(\cdot)$ and $k^{(n,b)}(\cdot)$ are respectively called descriptor of the background region, descriptor of the objects and descriptor of the boundary.

To compute an optimal solution, a dynamical scheme is introduced where the unknown regions become a function of an evolving parameter τ:

$$J_n(\tau) = \iint_{\Omega_{n,out}(\tau)} k^{(n,out)} + \iint_{\Omega_{n,in}(\tau)} k^{(n,in)} + \int_{\Gamma_n(\tau)} k^{(n,b)} \qquad (2)$$

Here $\Gamma_n(\tau)$ is modeled as an active contour that converges towards the final expected segmentation. Let Γ_{n_0} be the initial curve, we recall that we search for $\Gamma_n(\tau)$ as a curve evolving according to the following PDE:

$$\frac{\partial \Gamma_n(\tau)}{\partial \tau} = \boldsymbol{v_n} \qquad (3)$$

where $\boldsymbol{v_n}$ is the velocity of the active contour for frame number n. The main problem lies in finding the velocity $\boldsymbol{v_n}$ from the criterion (2) to get the fastest curve evolution towards the final segmentation.

2.2 The evolution equation of the active contour

In order to obtain the evolution equation, the criterion $J_n(\tau)$ must be differentiated with respect to τ. Let us define the functional $k_n(x, y, \tau)$ such that:

$$k_n(x,y,\tau) = \begin{cases} k^{(n,out)}(x,y,\tau) & \text{if } (x,y) \in \Omega_{n,out}(\tau) \\ k^{(n,in)}(x,y,\tau) & \text{if } (x,y) \in \Omega_{n,in}(\tau) \end{cases}$$

The criterion $J_n(\tau)$ writes as:

$$J_n(\tau) = \iint_{\Omega_n} k_n + \int_{\Gamma_n(\tau)} k^{(n,b)} = J_1(\tau) + J_2(\tau) \qquad (4)$$

where $\Omega_n = \Omega_{n,out}(\tau) \cup \Omega_{n,in}(\tau) \cup \Gamma_n(\tau)$.

In order to compute the derivative of the criterion J_1, discontinuities must explicitly be taken into account. The detailed proof can be found in [12], and we get the following expression:

$$J_1'(\tau) = \int_{\Gamma_n(\tau)} (k^{(n,out)} - k^{(n,in)})(\boldsymbol{v_n} \cdot \boldsymbol{N}) ds + \iint_{\Omega_{n,in}(\tau)} \frac{\partial k^{(n,in)}}{\partial \tau} dx dy + \iint_{\Omega_{n,out}(\tau)} \frac{\partial k^{(n,out)}}{\partial \tau} dx dy \qquad (5)$$

The derivative of J_2 is classical [4] and so, in the case of descriptors that do not depend on τ, the derivative of the whole criterion is the following:

$$J'(\tau) = \int_{\Gamma_n(\tau)} (k^{(n,out)} - k^{(n,in)} - k^{(n,b)} \cdot \kappa + \nabla k^{(n,b)} \cdot \boldsymbol{N})(\boldsymbol{v_n} \cdot \boldsymbol{N}) ds \qquad (6)$$

where $\kappa_n(x, y, \tau)$ is the curvature of $\Gamma_n(x, y, \tau)$.

For descriptors that do not depend on τ, according to the inequality of Cauchy-Schwartz, the fastest decrease of $J_n(\tau)$ is obtained by choosing $\boldsymbol{v_n} = F_n \boldsymbol{N}$, where:

$$F_n = [k^{(n,in)} - k^{(n,out)} + k^{(n,b)} \cdot \kappa - \nabla k^{(n,b)} \cdot \boldsymbol{N}] \qquad (7)$$

NB: For the case of time-dependent descriptors (descriptors depending on τ), the computation of the evolution equation is detailed in [12]. In this case, some additional terms appear in the velocity vector of the active contour. These terms have to be considered in order to get the fastest decrease of the active contour towards a minimum of the criterion.

In the case of time-independent descriptors, the curve evolves according to the following partial differential equation:

$$\frac{\partial \Gamma_n(\tau)}{\partial \tau} = [k^{(n,in)} - k^{(n,out)} + k^{(n,b)} \cdot \kappa - \nabla k^{(n,b)} \cdot \boldsymbol{N}]\boldsymbol{N} \quad (8)$$

In a video sequence, several objects may appear in a scene. So segmentation of video objects requires a method where topological changes are well handled in order to detect several objects from the same initial curve. On that account we use the level set method, proposed by Osher and Sethian [13], in order to implement the PDE (8).

3 VIDEO OBJECT SEGMENTATION WITH A 3-STEP ALGORITHM

The main goal of this part is to propose an algorithm to segment moving objects in video sequences without any user interaction during the segmentation. The user only has to choose a set of parameters at the beginning of the segmentation process.

Our 3-step algorithm successively operates a motion-based segmentation and two spatial-based segmentations, all of them using region-based active contours. We propose three sets of time-independent descriptors $\{k^{(n,out)}(\cdot), k^{(n,in)}(\cdot), k^{(n,b)}(\cdot)\}$ for each segmentation step. Obviously, the first set is motion-based while the two others are spatial-based.

As far as the motion-based step is concerned, we propose two options. The first one (a) is dedicated to video sequences with a static camera whereas the second one (b) is dedicated to sequences with a mobile camera.

The algorithm does not require initial object selection by the user. Indeed the initial contour is choosen to be a rectangle near the borders of the first image (see Fig.1). The contour is then driven by the first set of motion-based descriptors, which allows us to detect more or less precisely moving objects. We then use the final contour of this first detection as an initial curve for the second step. The second stage drives the active contour towards the nearest gradient of the image. The resulting contour is then taken as initial conditions for the third step which refines the detection by using segmentation of the image in intensity homogeneous regions. For each step, the active contour is driven by the PDE (8) by replacing the descriptors by their appropriate values.

NB: The color space used is (Y, C_b, C_r). The luminance Y of the image I_n is designated by $I_n(x, y, Y)$ while the two chrominances C_b and C_r are designated by respectively $I_n(x, y, C_b)$ and $I_n(x, y, C_r)$. In this article $I_n(x, y)$ designates $I_n(x, y, Y)$.

3.1 First step (option a): motion-based descriptors for a static camera

For a static camera, motion may be detected by comparing the current frame with a background frame B_n.

3.1.1 Computation of the background frame

In this paper, B_n is computed with a robust estimation on a group of frames including the current frame. This frame is not necessarily the real background of the sequence. We search for the frame B_n which minimizes:

$$\sum_{i \in [j, j+n_l]} \int_{\Omega_n} \varphi\left(|B_n - I_i|\right) \quad (9)$$

where n_l is the number of frames chosen by the user to compute B_n and j is a number of frame choosen such that $n \in [j, j + n_l]$.

In the Bayesian framework, φ is known as the potential function and is introduced to eliminate outliers. Here, we choose the Geman and Mc Lure estimator [14]:

$$\varphi(t) = \frac{t^2}{1 + t^2} \quad (10)$$

The minimization of (9) is done using the half quadratic theorem with the strategy based on alternate minimizations [5].

3.1.2 Motion-based descriptors for a static camera

The motion-based descriptors are thus the following:

$$\begin{cases} k^{(n,out)} = \sum_{\mathcal{V}} (B_n - I_n)^2 \\ k^{(n,in)} = \alpha_1 \\ k^{(n,b)} = \lambda_1 \end{cases} \quad (11)$$

where α_1 and λ_1 are two positive constants. The term $\mathcal{V}$ designates a neigbourhood of (x, y).

3.2 First step (option b): motion-based descriptors for a mobile camera

For a mobile camera, the idea is to assume that the apparent motion of the background can be modeled by a 6-parameter affine motion model. These parameters are computed with a robust estimation using motion vectors evaluated by a classical block matching. The moving objects are supposed to be the outliers of the robust estimation as well as pixels that are not compensated by the affine motion model. In this part, we first explain how the six parameters of the affine motion model are computed and then we detail the descriptors used.

3.2.1 Camera model

The camera motion can be modeled by a 6-parameter affine motion model which is a good trade-off between complexity and representativity [2]. So the apparent motion of a point (x, y) of the static background, between frames I_{n-1} and I_n, is modeled by:

$$\boldsymbol{w_n}(p) = \boldsymbol{A_n}\, p + \boldsymbol{t_n} = \begin{bmatrix} a_{11}^n x + a_{12}^n y + t_1^n \\ a_{21}^n x + a_{22}^n y + t_2^n \end{bmatrix} \quad (12)$$

with $p = (x, y)$.
We search for the six parameters of the camera model (12) which minimize the following criterion:

$$\sum_{p \in \Omega_n} \varphi\left(|\boldsymbol{u_n} - \boldsymbol{A_n}\, p - \boldsymbol{t_n}|\right) \quad (13)$$

The function φ is introduced to eliminate outliers due to the motion of moving video objects. We choose the Geman and Mc Lure estimator (10).

The motion field $\boldsymbol{u_n}$ is classically computed using a Block Matching algorithm between frames I_{n-1} and I_n.

In order to minimize the criterion (13), we use the properties of half quadratic regularization with the strategy based on alternate minimizations [5]. The initial minimization problem is in fact substituted for by the equivalent problem:

$$(\boldsymbol{A_n}, \boldsymbol{t_n}) = \text{argmin}_{(A_n, t_n)} \sum_{\Omega_n} b r^2 \quad (14)$$

where $r = |\boldsymbol{u_n} - \boldsymbol{A_n}\, p - \boldsymbol{t_n}|$ and $b = \frac{\varphi'(r)}{2r}$.

The algorithm based on alternate minimizations is then the following, with k the number of iteration:

Initialization $(\boldsymbol{A_n}^0, \boldsymbol{t_n}^0)$
Repeat

$b^{k+1} = \frac{\varphi'(r^k)}{2r^k}$

$(\boldsymbol{A_n}^{k+1}, \boldsymbol{t_n}^{k+1}) = \text{argmin}_{(A_n, t_n)} \sum_{\Omega_n} b^k (r^{k+1})^2$

Until convergence.

The minimization of $\sum_\Omega b^k (r^{k+1})^2$ is performed using a gradient descent method. We thus obtain the six parameters of the affine motion model.

At convergence of the algorithm, the image $b = \frac{\varphi'(r)}{2r}$ gives a representation of the outliers due to the motion of moving video objects. For a pixel (x, y), $b(x, y) \in [0, 1]$. The values near 0 are the outliers and so they correspond to the moving objects.

3.2.2 Motion-based descriptors for a mobile camera

The descriptors are then the following:

$$\begin{cases} k^{(n,out)} = \psi_1 \\ k^{(n,in)} = \frac{\varphi'(|\boldsymbol{u_n} - \boldsymbol{A_n}\, p - \boldsymbol{t_n}|)}{2|\boldsymbol{u_n} - \boldsymbol{A_n}\, p - \boldsymbol{t_n}|} + \frac{\varphi'(|I_n - Proj(I_{n-1})|)}{2|I_n - Proj(I_{n-1})|} \\ k^{(n,b)} = \lambda_1 \end{cases} \quad (15)$$

where λ_1 and ψ_1 are two positive constants.
The term $Proj(I_{n-1})$ designates the projection of the image I_{n-1} in the referential of image I_n:

$$Proj(I_{n-1})(p) = I_{n-1}(p + \boldsymbol{w_n}(p)) \quad (16)$$

3.3 Second step: gradient-based descriptors

Let Γ_n^1 be the final contour of step 1 (option a or b) and $\Omega_{n,in}^1$, $\Omega_{n,out}^1$ the two resulting domains. The region $\Omega_{n,in}^1$ contains pixels that are considered to belong to moving objects after step 1.
In step 2, we want to make the active contour evolve towards the nearest edges in the image, and so,we choose the following descriptors:

$$\begin{cases} k^{(n,out)} = c(|\nabla I_n|) \\ k^{(n,in)} = \alpha_2 \\ k^{(n,b)} = \lambda_2 \end{cases} \quad (17)$$

where λ_2 and α_2 are two positive constants.
The function c is defined as follows:

$$c(|\nabla I_n|) = \begin{cases} |\nabla I_n| & \text{if } (x, y) \in \Omega_{n,in}^1 \\ 0 & \text{otherwise} \end{cases}$$

This function allows to reach the nearest edge inside the first segmented region and not outside.

3.4 Third step: descriptors using regions partition

Let Γ_n^2 be the final contour of step 2, $\Omega_{n,in}^2$ contains the pixels of the image that are considered to belong to moving objects after step 2.

In this third step, we first make a partition of the image in intensity homogeneous regions using a region growing algorithm. We assume that a moving object is made with the union of several regions. We consider that if a region is mostly included in the final moving object of step 2, i.e. $\Omega_{n,in}^2$, then the active contour will be driven in order to include the whole region in the final moving object. On the contrary, if the region is mostly included in the background part, i.e. $\Omega_{n,out}^2$, then this region will be removed from the final segmentation.

3.4.1 Region growing method

Frames are segmented into intensity homogeneous regions. The pixel (x, y) belongs to the region R_i if it satisfies the following decision criterion:

$$|I_n(x, y, Y) - \mu_Y| \leq \sigma \text{ and } |I_n(x, y, C_b) - \mu_{C_b}| \leq \sigma \text{ and } |I_n(x, y, C_r) - \mu_{C_r}| \leq \sigma \quad (18)$$

where μ_Y, μ_{C_b}, μ_{C_r} are the average intensity values of the region R_i for respectively the luminance Y and the two chrominances C_b and C_r. The parameter σ is the variance, we take $\sigma = 8$.
We start from a pixel (x, y), and we check all the neighborhood points. Points that verify the criterion are inserted into the region. The region will expand until no

more neighborhood points can be added. The algorithm may be improved by using much more efficient methods such as binary partition trees [15].

3.4.2 Descriptors

The decision criterion used is based on the percentage of pixels included in $\Omega^2_{n,in}$. Let us call N_i the number of pixels of region R_i and $N_{i,seg}$ the number of pixels of region $R_i \cap \Omega^2_{n,in}$, ie the pixels of R_i that are inside the segmented region after step 2.
The descriptors are defined as follows:

$$\begin{cases} k^{(n,out)} &= d_{out} \\ k^{(n,in)} &= d_{in} \\ k^{(n,b)} &= \lambda_3 \end{cases} \quad (19)$$

where λ_3 is a positive constant.
The function d_{in} and d_{out} are chosen to be:

$$d_{in}(x,y) = \begin{cases} 0 & \text{if } \frac{|N_i - N_{i,seg}|}{Ni} \leq 0.02 \\ 1 & \text{otherwise} \end{cases} \text{ with } (x,y) \in R_i$$

$$d_{out}(x,y) = \begin{cases} 1 & \text{if } \frac{|N_i - N_{i,seg}|}{Ni} \leq 0.02 \\ 0 & \text{otherwise} \end{cases} \text{ with } (x,y) \in R_i$$

3.5 Remarks

The parameters α_i and ψ_i interact as threshold parameters, see [16] for detailed explanations. The parameter λ_i is a regularization parameter that smoothes the curve. In order to reduce the number of parameters to choose, we take $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$.

The final contour of frame number n is used as an initial contour for frame number $n+1$. The initial contour is re-initialized as a rectangle near the borders of the image every nb_{reinit} images, where nb_{reinit} is specified by the user (usually we take $4 \leq nb_{reinit} \leq 10$).

4 EXPERIMENTAL RESULTS

The algorithm has been tested on three real sequences, "Coastguard", "Mother and Daughter" and "Highway" from the research group COST 211 [17]. For Coastguard, we take $\psi_1 = 1.7$, $\lambda = 10$ and $\alpha_2 = 10000$. For "Mother and Daughter" we take $\alpha_1 = 80$, $\lambda = 15$, $n_l = 60$ and $\alpha_2 = 1000$, and for "Highway" we take $\alpha_1 = 400$, $\lambda = 15$, $n_l = 25$ and $\alpha_2 = 1000$.

In Fig.1, each step of the segmentation process is detailed. After the first stage, moving objects are roughly detected. The last two stages refine the result allowing an accurate detection of moving objects.

In Fig.2 and Fig.3, the final active contour is shown with a white envelop. Moving objects are well detected in both sequences either with camera motion or not, which illustrates the potential of our approach.

5 CONCLUSION

In this paper, we propose a 3-step algorithm to segment moving objects using region-based active contours. We propose 3 hierarchical stages to make the initial active contour evolve towards moving objects. The first step takes advantage of motion informations, while the two others take advantage of spatial informations, namely edges and intensity homogeneous regions of the image. This algorithm can be easily extended by adding new steps with some more informations as for example the temporal coherency of the video object.

6 REFERENCES

[1] International Organization for Standardization, "Overview of the MPEG-4 standard," March 1999, ISO/IEC JTC1/SC29/WG11 N2725.

[2] J. Odobez and P. Bouthemy, "Detection of multiple moving objects using multiscale MRF with camera motion compensation," in *ICIP*, Texas, 1994.

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Computer Vision*, vol. 1, pp. 321–332, 1988.

[4] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[5] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 298–311, february 1997.

[6] O. Amadieu, E. Debreuve, M. Barlaud, and G. Aubert, "Inward and outward curve evolution using level set method," in *ICIP*, Japan, 1999.

[7] S. Zhu and A. Yuille, "Region competition: unifying snakes, region growing, and bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 884–900, september 1996.

[8] A. Chakraborty, L. Staib, and J. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Trans. Medical Imag.*, vol. 15, pp. 859–870, 1996.

[9] A.R. Mansouri and J. Konrad, "Motion segmentation with level sets," in *ICIP*, Japan, 1999.

[10] N. Paragios and R. Deriche, "Geodesic active regions for motion estimation and tracking," in *Int. Conf. on Computer Vision*, Corfu Greece, 1999.

[11] C. Chesnaud, P. Réfrégier, and V. Boulet, "Statistical region snake-based segmentation adapted to different physical noise models," *IEEE Trans. PAMI*, vol. 21, pp. 1145–1156, nov. 1999.

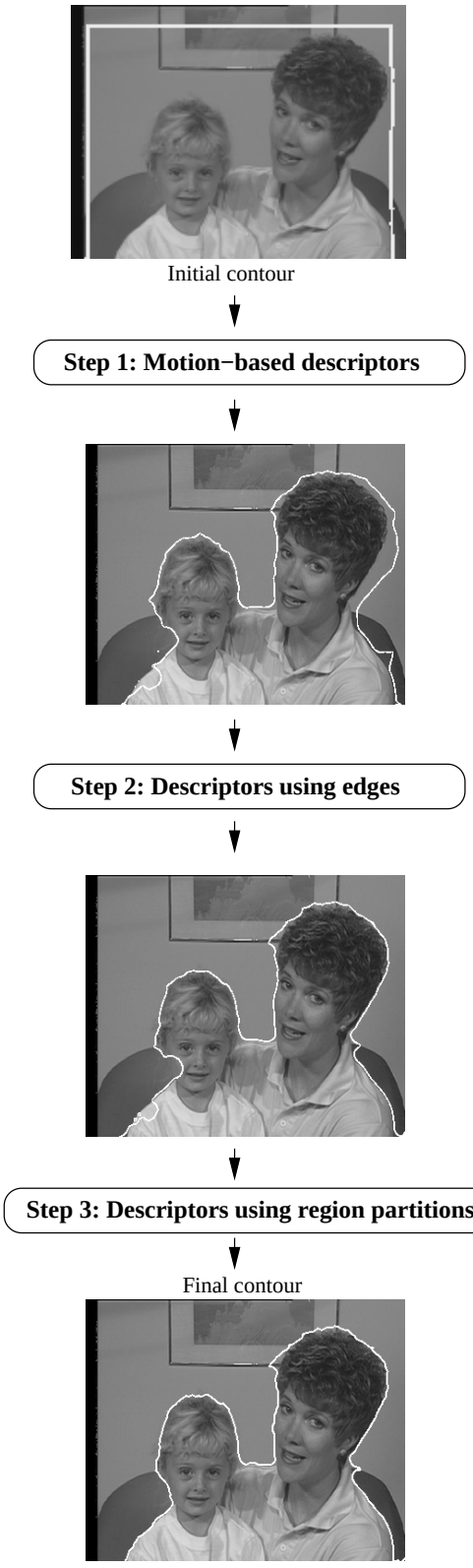


Figure 1: "Mother and Daughter", the 3-step algorithm

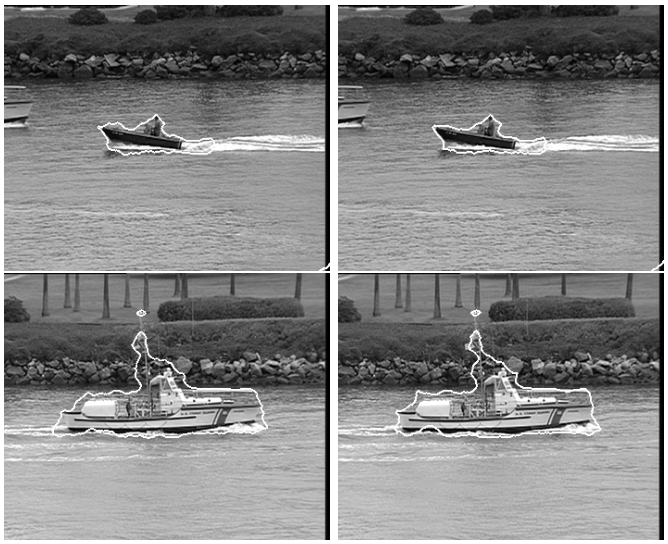


Figure 2: "Coastguard": Final contours in white

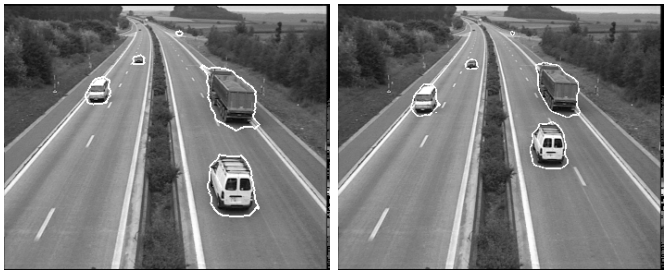


Figure 3: "Highway": Final contours in white

[12] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Region-based active contours for video object segmentation with camera compensation," in *Int. Conf. on Computer Vision*, Vancouver, 2001.

[13] J.A. Sethian, *Level Set Methods*, Cambridge Univ. Press, 1996.

[14] S. Geman and D.E. McClure, "Bayesian image analysis: an application to single photon emission tomography," in *Proc. Statist. Comput. Sect.*, 1985.

[15] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Trans. Image Proc.*, vol. 9, pp. 561–576, april 2000.

[16] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Detection and tracking of moving objects using a new level set based method," in *Int. Conf. on Pattern Recognition*, Barcelona, Spain, 2000.

[17] "http://www.tele.ucl.ac.be/exchange/," .

B-SPLINE ACTIVE CONTOUR FOR FAST VIDEO SEGMENTATION

F. Precioso and M. Barlaud
I3S laboratory - UPRES-A 6070 CNRS
Universit de Nice - Sophia Antipolis
2000 route des Lucioles - F-06903 Sophia-Antipolis FRANCE
precioso@i3s.unice.fr - barlaud@i3s.unice.fr

ABSTRACT

Video segmentation is among the most important problems of video processing and compression (standard MPEG-4 and MPEG-7). Unfortunately an important drawback for usual methods is the computation cost related to the model complexity. In this paper we propose to use a B-Splines parametric contour to implement a region-based active contours segmentation. Hence we get a fast variational method based on active contours with an intrinsic regularizing property. More precisely the evolution force is determined by minimizing a region-based criterion. The property of B-splines allows the computation of the contours curvature at any point using an analytic expression. The model complexity is fixed, depending on the desired level of detail, and is highly reduced as opposed to non parametric methods. Furthermore we compare this new approach to usual parametric polygon-based methods. We show experiments on a realisitic video sequence.

1 INTRODUCTION

Segmentation of moving objects in video sequences is a difficult and important problem in video processing.

Many approaches have been proposed to solve this problem. Two major kinds of technic exist. On one hand, there are statistical methods using Markov fields and discret models to model different parts of the scene [4]. On the other hand, some methods are based on variational approach and the minimization of a criterion. Most of these methods are based on active contours [2,5]. Level sets method are often chosen to implement methods based on active contours: Contour based on active contours [1], Region-based ones [3,6,9,10,13]. Obviously the advantage of level sets is an easier management of topological changes. The drawback is the high computation cost for such implementations. Contrarily parametric active contour methods can be fast method and efficient if the objects in the sequence do not undergo topology changes.

Thus, in this paper, we propose a parametric implementation of a region-based active contour approach. The aim is to obtain the segmentation with a contour evolution using a cubic B-spline interpolation [7,8,14]. Thanks to this interpolation the size of data in the algorithm is fully controled and highly reduced. Choosing cubic B-spline interpolation allow to preserve a C^2 regularity in each point of the contour. Hence terms depending on the contour's curvature can be computed. The principle of spatio-temporal segmentation method is a variational approach in order to minimize the following criterion :

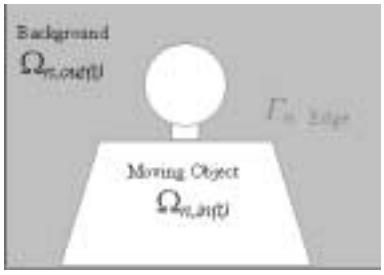


Figure 1: Domains definition

$$J_n(t) = \int_{\Omega_{n,out}(t)} k_{out} d\sigma + \int_{\Omega_{n,in}(t)} k_{in} d\sigma + \lambda \int_{\Gamma_n(t)} ds \qquad (1).$$

In this expression,t is the parameter for contour's evolution , n is the number of the current frame, $\Omega_{n,out}(t)$ represents the Background domain, $\Omega_{n,in}(t)$ represents the Object, and $\Gamma_n(t)$ the frontier between these domains.(**Fig.1**).

k_{out} is a background descriptor and k_{in} is a moving object descriptor. We currently use a temporal gradient $k_{out} = (S_n - S_{n-1})^2$ with S_n the frame n in the sequence and $k_{in} = \alpha_c$ with α_c a constant.

By distributions differentiation on the criterion (1), we obtain a *Propagation-PDE* representing the contour evolution:

$$\begin{cases} \frac{\partial \Gamma_n(t)}{\partial t} = F\vec{N} = \vec{V} \\ \Gamma_n(0) = \Gamma_{n,0} \end{cases} \qquad (2).$$

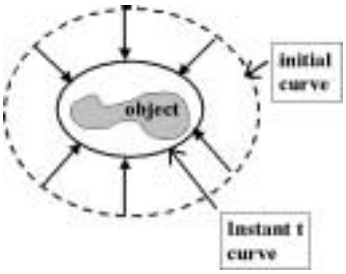


Figure 2: Contour Evolution

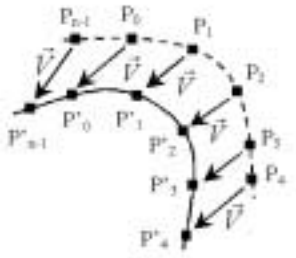


Figure 3: Interpolated Propagation

From an initial curve $\Gamma_n(0)$ given, the contour $\Gamma_n(t)$ evolve thanks to velocity definition : $\vec{V} = F\vec{N}$, following the normal direction of the contour.

The expression of the force is deduced from the criterion (1) [3,13]:

$$F = k_{in} - k_{out} + \lambda\kappa \qquad (3).$$

with κ the contour curvature of the contour (**Fig.2**).

One might implement active contour evolution following two different ways:

– Implicit implementation : A really efficient "level set"-based method already exist to implement video segmentation from this criterion (1) [3,13]. Its main advantage is to provide an implicit management of topological changes. On the other hand computation cost of this approach is important.

– Explicit implementation : The purpose of this paper is to use parametric active contours. Two straight advantages come out: A low computation cost because of data size to process and a total control of this computation cost thanks to our knowledge of the number of evolution points to manipulate. A second advantage of this method lies in manipulating analytic expressions.

Usual parametric acitve contour technics deal with polygons. However that kind of approach request lot of points in order to keep a satisfying regularity.

2 PROPAGATION METHOD

2.1 Principle of the method

In this paper we propose to use B-Spline curves (curve composed by several fixed degree Bézier curves) because these curves are C^2 regular in each point of the contour. Our goal is to reduce the number of points and thus to reduce computation cost.

The principle is based on applying the force (3), over few contour points. Then contour evolution depends only on the evolution of these interpolated points and no more on each point of the contour. Using B-splines allow to preserve the C^2 continuity. Thus the third term of force F (3), a regularization term over the contour length (depending on the curvature κ), is defined in each point.

Our algorithm is composed of three steps:
- Active contour interpolation
- Curvature computation κ eq.(7)
- Propagation eq.(2)

2.2 Active contour interpolation

We determinate a contour (ellipse, circle, ...). Then this *fixed* contour is interpolated regularly.

P_n represent contour interpolated points which are going to evolve.

After calculating the curvature κ in each point P_n, the evolution force (3) is determined in these points through criterion minimization and evaluated with the new curvature, and each descriptor of objects and background.

After a propagation step, we obtain points $P_k^{'}$ from points P_k (as shown on figure **Fig.3**).

The contour interpolation is based on cubic Uniform B-Spline curves. The uniformity is employed here to improve computation efficiency as we are going to see below. Third degree B-Splines are used because we only need second derivation term in the curvature expression (7).

A cubic B-Spline curve is given by a polynomial expression as below[7]:

$$\begin{aligned} S_i(s) = {} & Q_{i-1}B_{S^4_{i-3}}(s) + Q_iB_{S^4_{i-2}}(s) \\ & + Q_{i+1}B_{S^4_{i-1}}(s) + Q_{i+2}B_{S^4_i}(s). \end{aligned} \qquad (4).$$

s is the parameter of the curve (in fact curvilinear abscise), points Q are "anchors" of the B-Spline (called usually: Control Points), and $B_{S^4_i}(s)$ are polynomial expressions (with 2 components) defining basic functions of the B-Spline(hence "Basic-Spline"), it means weight functions of each point for the arc $S_i(s)$.

It has to be noticed that in all these expressions Q and P represent points in a frame of the sequence. So they both have 2 components. It's the same with $S_i(s)$ which is the parametric equation of the arc between P_i and P_{i+1}. Thus P_i represent $(x_i(s), y_i(s))$ with $P_i =$

$(x_i(s_i), y_i(s_i))$ and $P_{i+1} = (x_i(s_{i+1}), y_i(s_{i+1}))$. The arc equation is [8]:

$$S_i(s) = \begin{pmatrix} 1 & s & s^2 & s^3 \end{pmatrix} \begin{pmatrix} & M_i & \end{pmatrix} \begin{pmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{pmatrix}$$

in which M_i matrix represent the polynomial expression coefficients of the B-Spline arc between P_i and P_{i+1}.

If all couples (P_i, P_{i+1}) are supposed regularly distributed along the curve, $\forall i \in [0, .., n-2]$ $\triangle s_i = \|s_{i+1} - s_i\|$, a new parametrization for each arc into $[0, 1]$ leads to easier computation. Basics functions $B_{S_i^4}(s)$ are identical for all arcs as we are using a *Uniform* B-Splines interpolation.

Using Uniform B-Spline and after parametrizing each arc into $[0, 1]$, M_i matrix are identical for all arcs ($\forall i \in [0, .., n-1]$). This unique matrix called M from now is definied for one arc of Uniform B-Spline by [8]:

$$\forall i, M_i = M = \frac{1}{6} \begin{pmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix}$$

Each interpolated point P_i corresponds to the polynom $S_i(s)$ value when $s = s_i$. By using s_i into Uniform B-Spline arc between P_i and P_{i+1} (4) expression we obtain the relation between n interpolated points $P_i, i \in [0, ..., n-1]$ and $n+2$ arc control points, $Q_i, i \in [-1, ..., n]$:

$$S_i(s_i) = P_i = \frac{1}{6}(Q_{i-1} + 4Q_i + Q_{i+1}) \qquad (5).$$

Number of control points should be $n+2$ but as we want to obtain a closed curve as result, we use twice the first and the last point Q_0 and Q_{n-1}. So we have:

$$Q_{-1} = Q_{n-1}$$

$$Q_n = Q_O$$

Thus to obtain control points Q of the Uniform B-Spline curve from interpolated points P . We only need to compute the inversion of a circulating $n \times n$ matrix.

$$\begin{pmatrix} P_0 \\ P_1 \\ \dots \\ P_{n-2} \\ P_{n-1} \end{pmatrix} = \begin{pmatrix} 4 & 1 & 0 & \dots & 0 & 1 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ 0 & \dots & 0 & 1 & 4 & 1 \\ 1 & 0 & \dots & 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ \dots \\ Q_{n-2} \\ Q_{n-1} \end{pmatrix}$$

(6).

Thanks to proprieties of these matrices and to improve our algorithm, this inversion is based on Fourier Transformation: Naming p_n interpolated points vector, q_n control points vector, and h_n Fourrier coefficients vector:

$$h_n = \begin{pmatrix} 4,1,0,\dots,0,1 \end{pmatrix}^T$$

Assuming q_n and h_n represent part of complex periodic sets $(q_n)_{n\in\mathbb{Z}}$ and $(h_n)_{n\in\mathbb{Z}}$, with period n, we can consider p_n as a part of the periodic set $(p_n)_{n\in\mathbb{Z}}$ convolution of previous ones:

$$\forall n \in \mathbb{Z} \qquad p_n = \sum_{k=0}^{n-1} h_{n-k} q_k$$

This expression can become using the linear system (6) as:

$$p_n = Hqn$$

But then we can name $\mathcal{P}_k$,$\mathcal{Q}_k$ and $\mathcal{H}_k$, Discret Fourrier transform of $(p_n)_{n\in\mathbb{Z}}$,(q_n) and $(h_n)_{n\in\mathbb{Z}}$. Hence the previous linear system is turned into:

$$\mathcal{P}_k = n\mathcal{H}_k \mathcal{Q}_k \qquad k = 0, 1, ..., n-1$$

This computation concern only complex n-long vectors. So $\mathcal{Q}_k$ and by reverse Discret Fourrier transform, q_n is known.

It has to be noticed that these calculus depend only on the number of interpolated points.

2.3 Curvature equation

Thanks to B-Spline curves and their fundamental propriety of C^2 regularity in each point of the curve (even joining "segments" points), we can evaluate the third term of the criterion (1) which concern curvature of the contour.

We compute at each one of interpolated points P_k the curvature $\kappa_k(s)$. We use coefficients of polynomial expressions of each $S_i(s)$ components to obtain the curvature through the relation:

$$\kappa_i(s) = \frac{x_i'(s)y_i''(s) - x_i''(s)y_i'(s)}{\left(x_i'(s)^2 + y_i'(s)^2\right)^{\frac{3}{2}}} \qquad (7).$$

with $x_i(s)$ and $y_i(s)$ the first component and the second components of $S_i(s)$ respectively. .

2.4 Propagation

So we have P_k^0 points by initializing (step 0) along a determined contour.(As the contour is already determined we do not need to use anchors, so we do not need to control points for now). After applying the force on each point, we have new points $P_k^{'}$ (step 1). From these points we determine first control points Q_k^1 (step 1).

At each iteration we compute coefficients of each arc of the Uniform B-Spline interpolating the active contour. These arcs are defined with cubic polynomial expressions. Therefore results are two analytic expressions for each arc (one per component) and only coefficients of these expressions have to be stored.

The new force F value is computed in each point of P_k^1 using the curvature value (7) (step 3) .This computation is based on analytic expressions of arcs composing the contour. This results in a faster and easier than using the level set method.

We consider interpolated points P, evenly sampled along the curve. This approximation allows us to interpolate the curve using points P_k into Uniform B-Spline interpolation. As we delete or create new interpolating points with a threeshold as in some polygon based methods [12], the approximation of regular sampling is not absurd. Moreover experimantal results confirm it.

Then we determine control points (anchors of the curve), points Q, for each arc linking P previous points.

Once we know new positions of P_k^1 points we can immediatly evaluate the expression of the curvature κ (step 2) all along the curve and then the new value for the force F at these points.

This computation is repeated until convergence. Through studies on the criterion (1) [3,13], we know this evolve towards moving objects. So we consider convergence realized with the end of evolution of the contour.

3 EXPERIMENTS and COMPARISON

Fig.4 shows the contour initialization for segmentation of a moving speacher in a video sequence. **Fig.5**) is the result of the convergence of our algorithm. We obtain efficient results extracting the moving object from a video sequence.

Moreover we can compare our method with classical polygon-based ones:

In our algorithm, we use an analytic formula to compute the curvature (7). As we assume interpolated points P are evenly spaced, the parameter t of the curve evolve from 0 to 1 on each cubic segment composing the curve. Hence to compute the exact value of the curvature at the point P_i, we just use:

$$\kappa_i = \kappa_i(0) = \frac{x_i'(0)y_i''(0) - x_i''(0)y_i'(0)}{\left(x_i'(0)^2 + y_i'(0)^2\right)^{\frac{3}{2}}}$$

where each of $x_i'(0)$, $x_i''(0)$, $y_i'(0)$, $y_i''(0)$ is a float, factor from the polynomial expression $S_i(s)$ with $t = 0$. This calculus is independant on the point considered (no needs of a specific management for first and last interpolated points).

An other advantage in using analytic expressions is to allow direct computation of normal vector. In our algorithm the normal vector at P_i is:

$$\vec{N_i} = \begin{pmatrix} \frac{-y_i'(0)}{\sqrt{x_i'(0)^2+y_i'(0)^2}} \\ \frac{x_i'(0)}{\sqrt{x_i'(0)^2+y_i'(0)^2}} \end{pmatrix}$$

Computation cost per interpolating point with B-Spline-based method:

- $\kappa_{B-Spline}$: 8 operations/point.
- $\vec{N}_{B-Spline}$: 12 operations/point.

Polygon-based methods use as approximation for the curvature at P_i, the length of the median in the triangle (P_{i-1},P_i,P_{i+1}) [11].

Once again, polygon-based methods need an approximation [12]. Since curvature does not exist for polygons,

Figure 4: Interpolated Contour Intialization
Interpolation using 100 points, $k_{out} = (S_n - S_{n-1})^2$, α_c= 10, λ = 10

Figure 5: Interpolated Contour Convergence

we define the normal vector as the opposite of the average vector $\vec{N}$ from normal vectors of $\vec{N_1}$ previous and $\vec{N_2}$ of next edge of both side of the point P_i (as shown on **Fig.6**).

Computation cost per interpolating point with polygon-based method:

- $\kappa_{Polygon}$: 13 operations/point.
- $\vec{N}_{Polygon}$: 28 operations/point.

The computation cost comparison is recapitulated here below:

	κ	$\vec{N}$	Total
Spline	8 op./p.	12 op./p.	20 op./p.
Polygon	13 op./p.	28 op./p.	41 op./p.

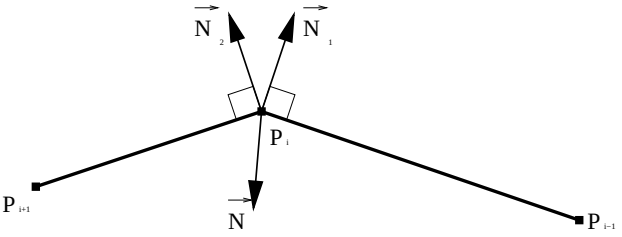


Figure 6: Normal vector approximation for Polygon-based method

These calculus have to be computed at each interpolated point of the contour. The simpler they are, the faster the algorithm will be. Thanks to the C^2 continuity of B-Splines, less points are needed to interpolate the active contour than with polygon curves. Cubic B-Splines provide a highly regular contour. However, the contour is composed by cubic B-Spline "segments". Thus we controlthe curve parameters at a local scale.

4 CONCLUSION

In this paper, we proposed to use a new active contours method using B-Splines interpolation to implement a region-based active contours segmentation algorithm. Our goal was to preserve the segmentation quality while reducing computation cost compared to level set methods. Model deformations are controled by criterion (1), including a curvature term, which is known to produce good results results for video segmentation [3,13]. Quality of results is satisfying as illustrated in **Fig.5**. The number of operations computed at each point is lower than usual polygon methods thanks to B-Spline analytic expressions. The property of cubic B-Splines C^2 continuity enforces contours regularity. The contour being assembled from individual segments, each property such as regularity or precision can be controled at a local scale, segment by segment.

References

[1] J-M. Odebez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models", *J. Visual Communic. And Image Repres.*, 1995, 6(4), pp. 348-365.

[2] V. Caselles, R. Kimmel and G. Sapiro, "Geodesic active contours", *International Journal of Computer Vision*, 22(1), pp. 61-79, 1997.

[3] S. Jehan Besson, M. Barlaud and G. Aubert, "Detection and Tracking of Moving Objetcs Using a New Level Set Based Method", ICPR 2000.

[4] A-R. Mansouri and J. Konrad, "Motion Segmentation with Level Sets", ICIP, 1999, Kobe.

[5] M. Kass,A. Witkin and D. Terzopoulos, "Snakes: Active contour models", *International Journal of Computer Vision*, vol. 1, pp. 321-332, 1987.

[6] N. Paragios and R. Deriche, "Geodesic active regions for motion estimation and tracking". IEEE ICCV, Corfu, Grèce, 1999.

[7] R. H. Bartels, J. C. Beatty, B. A. Barsky, "*An introduction to Splines For use in Computer Graphics and Geometric Modeling.*" Morgan-Kaufmann, Los Altos, Calif., 1987.

[8] James D. Foley, Andries Van Dam, Steven K. Feiner and John F. Hughes (1990a). "*Computer Graphics Principles and Practice*". The Systems Programming Series. Addison-Wesley Publishing Company, Inc.

[9] C. Zhu et A. Yuille, "Region competition: Unifying snakes, region growing and Bayes/MDL for multiband image segmentation", IEEE PAMI, vol. 18, n 9, pp. 884-900,1996.

[10] A. Chakraborty,J. S. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information", IEEE Trans. Medical Imaging, 1996, 859-870.

[11] C. Chesnaud, P. Refregier, V. Boulet, "Statistical Region Snake-Based Segmentation Adapted to Different Physical Noise Models", IEEE PAMI, vol. 21, n 11, 1999.

[12] M. Maziere, F. Chassaing, "Segmentation and Tracking of video objects: suited to content-based video indexing, interactive television and production systems", ICIP 2000, sept., Vancouver.

[13] S. Jehan, E. Debreuve, M. Barlaud, G. Aubert, "Segmentation spatio-temporelle d'objets en mouvement dans une squence vido par contours actifs dformables", RFIA, Paris, fvrier 2000.

ROBUST CONTENT BASED IMAGE WATERMARKING

Selena Kay and Ebroul Izquierdo

Multimedia&Vision Research Lab
Departement of Electronic Engineering
Queen Mary, University of London, London, U.K
Email: {ebroul.izquierdo, selena.kay}@elec.qmw.ac.uk

ABSTRACT

A watermarking scheme is presented in which characteristics of both spatial and frequency techniques are combined to achieve robustness against image processing and geometric transformations. The proposed approach consists of three basic steps: estimation of the just noticeable image distortion, watermark embedding by adaptive spreading of the watermark signal in the frequency domain, and extraction of relevant information relating to the spatial distribution of pixels in the original image. The just noticeable image distortion is used to insert a pseudo-random signal such that its amplitude is maintained below the distortion sensitivity of the pixel into which it is embedded. Embedding the watermark in the frequency domain guarantees robustness against compression performed in image processing attacks. In the spatial domain most salient image points are characterized using first order differential invariants. This information is used to detect geometrical attacks in a frequency-domain watermarked image and to re-synchronize the attacked image. The presented schema has been evaluated experimentally. The obtained results show that the technique is resilient to most common attacks including geometrical image transformations.

1. INTRODUCTION

Conventional analog media distribution systems have an inherent built-in defense against copying, alteration and fraud. Each time a new copy is issued the quality of the duplicated content is degraded accordingly. In contrast to that, digital multimedia documents are completely susceptible to exact replication and alteration. This, together with the rapid proliferation of digital documents, multimedia processing tools and the world-wide availability of internet access have created an ideal medium for piracy, copyright fraud and uncontrollable distribution of high quality but unregistered multimedia content. Since digital watermarking can be seen as a solution to this problem, both the number of activities in this area and the recognition of the difficulties and challenges involved in this new technology has increased in the last few years [1-2].

Usually digital watermarks are classified according to the embedding and retrieval domain, i.e. the luminance intensity in the spatial domain and the transform coefficient magnitude in the frequency domain. Frequency based techniques, are very robust against certain kinds of transformations, such as compression and filtering. Since the watermark is spread throughout the image data, rather than targeting individual pixels, any attempts at attack means that the most fundamental structural components of the data must be targeted. In this context many watermarking algorithms relying on Spread Spectrum have been proposed in the literature [3-4]. However, reliable detection of the frequency-based watermark is impeded when synchronization is lost as a result of geometric transformations. To deal with these attacks a spatial watermark is more appropriate as it targets specific locations in the image. Watermarking in the spatial domain is less resilient to common image processing operations, since the watermark becomes undetectable when the intensity information is modified. However, spatial techniques allow relevant information relating to the spatial distribution of pixels to be extracted. With this information, following geometric transformations such as rotation, the image can be resynchronized. The most common strategy for detecting a watermark after geometric distortion is to try to identify what the distortions are and then to invert them before applying the detector, e.g. by introducing a template [5-6]. This requires the insertion and the detection of two watermarks: one of which does not carry information but which helps to detect geometric transformations and a second one in which the hidden information is represented. This approach has two drawbacks: it further affects image fidelity and it increases the probability of false negatives. Additionally, in general this technique requires exhaustive searches thus resulting in a significant increase of workload. Since the watermarking strategy should be public it is rather easy to destroy the synchronization template.

In this paper an imperceptible and robust watermarking scheme for still images is described. The proposed technique consists of three basic steps: 1) content based estimation of *just noticeable distortion* (JND) in the frequency domain; 2) adaptive spreading of a pseudo-random watermark signal in the frequency domain; and 3) extracting relevant information relating to the spatial distribution of pixels in the original image in order to re-synchronize an image distorted by a geometrical attack. In

the first step the image is analyzed in both the frequency and the spatial domain in order to detect the distortion sensitivity of the image according to its content. Local information derived from texture, edge and luminance information is used to define a measure of JND. The JND is used adaptively according to the image content to maximize the amount of information (signal) that will be embedded as the watermark. To insert the watermark signal the middle DCT-frequency band of a block-wise transformed image is used. To maximize the capacity a pseudo-random signal with amplitude just below the image distortion sensitivity is created according to the JND mask. Thus, the watermark signal is spread over the whole host by keeping its amplitude below the noise sensitivity of each pixel.

Information extracted from the spatial domain is used to resynchronize the image in the case of geometric attacks. This is implemented by applying primitive matching using point characterisation according to differential invariants [7]. These invariants have been defined in the literature for grey-level images. For colour images this characterisation can be improved if the three colour channels are used. In this case only first order invariants are necessary. Using the colour information achieves a more efficient characterrisation as only first order derivatives are used. In this work the Harris corner detector [8] is used to detect salient image points. First order Hilbert invariants [7] are selected to characterize the salient image primitives. To detect the transformations undergone by the attacked image a matching technique is used.

The paper is organised as follows: In section 2 the method for extraction of the JND mask is described. Section 3 deals with watermark insertion and extraction in the DCT domain. Correction of geometric distortions using Hilbert invariants is described in section 4. The paper closes with a brief report on selected results of computer simulations in section 5.

2. GENERATION OF THE JND-MASK

The process of embedding a watermark in any image can be regarded in the same way as adding noise to the image. This process leads to an alteration of the host image. Obviously altering a large number of pixel values arbitrarily will result in noticeable image distortions. These distortions are proportional to the amplitude of the embedded signal. Consequently, an image can be distorted only to a certain limit without making the difference between the original and the altered one perceptible. This limit varies according to the image content and is called *just noticeable distortion* (JND). To estimate the JND mask three image characteristics are considered: texture, edgeness and smoothness. According to several studies about the human vision system, it is well-known that the distortion visibility in highly textured areas is very low. This means that these areas are the most suitable ones in which to hide the watermark signal. In contrast to that, edge information of an image is the most important factor for the human vision perception. Consequently, edges have the lowest just noticeable distortion values. Similarly, smooth image areas have a general bandpass characteristic. They influence human perception and consequently their JND values are also relatively low. The definition of a suitable JND mask depends essentially on the accurate extraction of image texture, edges and smooth areas.

Texture information can be retrieved directly from the transformed domain by analyzing the DCT coefficients. Following the JPEG and MPEG2 encoding strategy, the image is first split into 8X8 blocks. Each block is transformed in the DCT domain and the resulting 64 coefficients analyzed. It is well known that in highly textured regions or along edges the signal energy is concentrated in the high frequency coefficients while in uniform image areas the signal energy is concentrated in the low frequency components. Using the following formula for the energy in the AC coefficients a measure for texture D_T within each block is derived:

$$D_T = \log\left(\sum_{i=1}^{63} v_i^2 - v_0^2 \right), \qquad (1)$$

where $v_i,\ i = 0,...,63$ are the 64 DCT coefficients of the considered block. The obtained D_T values are scaled to the range [0, 64] and the resulting normalized values assigned to the corresponding blocks.

Edges and smooth areas are extracted from the pixel domain. The main difficulty here consists of discriminating between relevant image edges and spurious edges due to noise and texture. To this aim the classic Gaussian scale-space model is considered. Main edges are first identified regardless of their position at low scales. Then they are shifted to their correct position using decreasing variance values. The edges are detected using either the Canny operator or by extracting zero-crossings of the LOG operator. The length of each single edge is calculated by traversing it. Edges whose length does not exceed a threshold are considered texture edges and are removed. Using the binary image containing the final edge information, edgeness is calculated block-wise according to the formula:

$$D_E = \frac{64.P_E}{\max(P_E)}, \qquad (2)$$

where P_E is the cardinality of the set of pixels within the block and at edge locations and $\max(P_E)$ is the maximum value of P_E over all the blocks in the image.

Finally, the Moravec operator is used to extract uniform image regions. The same strategy introduced in [9] for the recognition of uniform regions is used. The uniformity D_U in a block is defined as the number of pixels belonging to a uniform area within the block.

Using the three values D_T, D_E and D_U an initial estimate for the *JND* value over each block is calculated as: $\tilde{J} = D_T - \frac{1}{2}(D_E + D_U)$. Since the human vision system is more sensitive to intensity changes in the mid-gray

region and its sensitivity fail parabolically at the both ends of the gray scale a correction to $\tilde{J}$ is introduced. The final *JND* values are obtained as:

$$J = \tilde{J} + \left(128 - \tilde{I}\right)^2, \quad (3)$$

where $\tilde{I}$ is the average of the luminance values within the considered block.

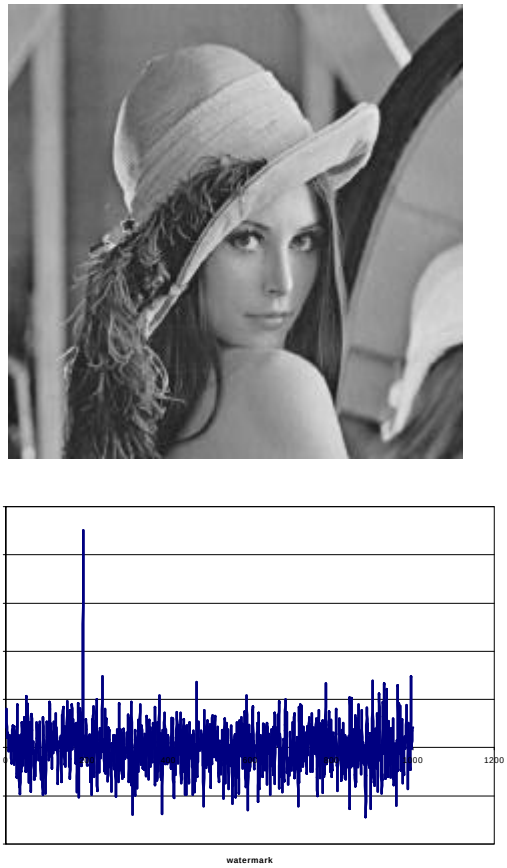


Fig. 1: Watermarked Lena and response of the watermark detector after JPEG-compression using 50% quality factor.

3. WATERMARK EMBEDDING IN THE DCT DOMAIN

The method of watermark insertion is to modify selected DCT coefficients by embedding a sequence of pseudo-random numbers within them. This technique does not target individual pixels but rather, upon inverse transformation, the watermark will be dispersed over the entire image. The watermark itself consists of a sequence of real pseudo-random numbers $X = x_1,...,x_m$, with $m \le 64$, satisfying the normal distribution $h(0,1)$. The array of DCT coefficients obtained from the transformation consists of a sequence of values $V = v_1,...,v_{64}$. The watermark $X = x_1,...,x_m$ is inserted into m selected coefficients from the middle frequency bands. This yields an adjusted set of values $V' = v'_1,...,v'_{64}$. The inverse of the DCT is then performed to obtain the watermarked image I'. To insert the watermark we use the formula

$$v'_i = v_i + a.J.|v_i|x_i \quad (4)$$

where a is a scaling parameter and J is the distortion parameter defined by (3) and corresponding to the block concerned. If the watermarked image I' is transformed, by processing or attack, a new image I^* is generated. The presence of the watermark can be detected in the transformed image I^* by performing a correlation test.

$$R = \frac{1}{N}(V^* \cdot X) \quad (5)$$

where V^* is the vector containing those extracted DCT coefficients which have been modified and X is the original watermark. Since this is a statistical test we must be aware of the possibility of obtaining detection errors. To decide whether the watermark is authentic we must determine some threshold T and test whether the obtained correlation coefficient is greater than T. Setting the detection threshold is a decision based on the desire to minimise both false alarms and false rejections.

4. DETECTION OF GEOMETRIC ATTACKS

Since most digital images are colour rather then grey-level the entire colour information can be exploited using well-known grey-level image attributes independently for each colour plane. We are interested in attributes that are invariant with respect to as large a group of geometric transformations as possible, but specifically in orthogonal and affine transformations. The study of these invariants can be traced back to the 18th century in work undertaken by the renowned mathematician David Hilbert [8]. He showed that any invariant of finite order can be expressed as a polynomial function of a set of irreducible invariants. In a grey-level image these fundamental set can be defined as: I , I_h , I_{hh} , I_{Vh} , I_{VV} , with the unit vector h given by $h = \frac{\nabla I}{|\nabla I|}$ and $V \perp h$.

Given an original image I the first step in generating a feature space with invariant attributes is to detect some key points that are robust to common image processing and geometrical transformations. We have chosen the Harris detector which uses only first order derivatives and is well known as one of the most stable and robust corner detectors in image processing. The Harris detector is defined as the positive local extreme of the following operator: $Det(M) - kTrace^2(M)$, where $k=0.04$ is a scalar and the matrix M is given by:

$$M = \begin{bmatrix} a = \Gamma s\,(R_x^2 + G_x^2 + B_x^2) & c \\ c & b = \Gamma s\,(R_y^2 + G_y^2 + B_y^2) \end{bmatrix},$$

with $c = \Gamma s\,(R_x \cdot R_y + G_x \cdot G_y + B_x \cdot B_y)$ and Γ the Gaussian convolution with a kernel of variance s .

As mentioned before the basic idea behind the strategy for defining an invariant feature space is to use first order

Hilbert invariants. For each corner point the vector comprising the following eight differential primitives is calculated:

$$F = (R, |\nabla R|^2, G, |\nabla G|^2, B, |\nabla B|^2, \nabla R \cdot \nabla G, \nabla R \cdot \nabla B)^T \quad (6)$$

The vector (6) forms the space of feature invariants in the considered colour image. It is invariant with respect to rotation, translation and scaling which are the most common geometric transformations. It allows very robust characterisation with regard to noise, since only first order derivatives are involved. Additionally, the complexity of the method remains very low since only simple pixel differences should be calculated. The use of higher order derivatives involves heavier workload and would cause more instability.

Fig. 2: Selected relevant corners for re-synchronization.

5. EXPERIMENTAL RESULTS

Several experiments have been conducted to test the performance of the proposed strategy. In this section some selected results are reported. The image at the top of Fig. 1 shows the watermarked image Lena using the technique described in section 3. The image at the bottom shows the response of the watermark detector after JPEG compression using 50% quality factor. Tests against geometric attacks such as cropping, rotations, scaling, etc., have been carried out. In most cases good synchronization has been achieved. For cropping attacks the image must be padded to the original size before decoding. Thus frequency sampling is performed with the same step both by the encoder and the decoder. In this case cropping robustness is obtained from the invariance of the magnitude spectrum to spatial shifting. In all cases the parameters describing the performed geometrical attack have been detected using the technique described in section 4. Fig. 2 shows the detected corners in the watermarked image. In this representation corners are highlighted as small black spots. Matching between image primitives using the Hilbert invariant features described in section 4 reveal the parameters encoding the geometric transformations undergone by the attacked image. These parameters have been used to re-synchronize the transformed image and to detect the watermark in the DCT domain. Fig. 3 shows the response of the watermark detector for a rotated image. In this case the image shown in Fig. 1 has been rotated 10 degrees to the left. Using the technique described in section 4 the rotation angle has been estimated and the attacked image rotated back. The watermark detector has been applied to the re-synchronized image.

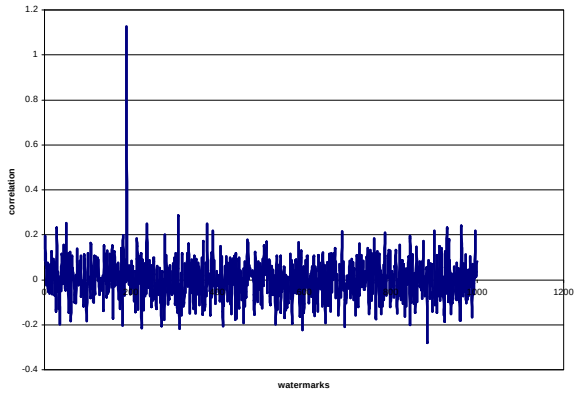


Fig. 3: Response of the watermark detector after re-synchronization.

References

[1] *Signal Processing*, Special Issue on Watermarking, vol. 66, no. 3 May 1998.

[2] Proc. IEEE, Special Issue on Identification and Protection of Multimedia Information, vol 87, no. 7, Jul. 1999.

[3] Chiou-Ting-Hsu, and Ja-Ling-Wu, "Hidden digital watermarks in images", in *IEEE Transactions on Image Processing*, vol.8, no.1, pp. *58-68*, 1999.

[4] R.B. Wolfgang, C. Podilchuk., and, E. J. Deip, "Perceptual Watermarks for digital images and video", *Proc. of the IEEE*, vol.87, no.7, pp. 1108-1126, 1999.

[5] S. Pereira and T. Pun, "Fast robust template matching for affine resistant image watermarks' in *Proc. of the 3rd internat. Information Hiding Workshop*, 1999, pp. 207-218.

[6] M. Kutter, "Watermarking resistance to translation, rotation, and scaling' in *SPIE Conf on Multimedia systems and Applications*, 1998, vol. SPIE *3528*, pp.423-431.

[7] D. Hilbert, "Theory of Algebraic Invariants" Cambridge Mathematica Library, Cambridge University Press, Cambridge, 1890.

[8] C. Harris, M Stephens, "A combined corner and edge detector", Proceedings of the fourth Alvey vision conference, 1988, pp. 147-151.

[9] E. Izquierdo, "Stereo Image Analysis for Multi Viewpoint Teleprecense Applications", *Signal Processing: Image Communication*, Vol. 11, No. 3, Jan. 1998, pp. 231-254.

A Platform for Multi-Service Residential Networks

Eric M. Scharf
(On behalf of the TORRENT Project)
Department of Electronic Engineering
Queen Mary, University of London

Abstract

The aim of the EU-Supported Framework V project TORRENT is to build a test-bed for multi-service residential networks. This test-bed will allow the project to demonstrate the benefit of intelligent control, both for the customer and for the network operators and service providers. This intelligence will enable a home user's QoS expectations for particular range and combination of services, to be met, in a transparent way, by the ability to choose the most appropriate core transport network and physical access interfaces.

Introduction

Connection-oriented and connectionless networks have evolved separately, since each has characteristics making it more suitable for particular types of services. Connection-oriented networks have typically been better suited to services characterised by long holding times, few (e.g. two) parties in the call, low congestion tolerance, and often, high QoS needs and large data content. Connectionless networks, on the other hand, are associated with services of a multi-cast or broadcast nature, and services that are characterised by high tolerance to congestion, non-critical QoS needs and low data content.

Equipment manufacturers supporting these two different kinds of networks are continuously improving their respective technologies with the aim of demonstrating that their type of network technology is the ideal candidate for a future single, multi-service, integrated broadband network. However, it is by no means certain that a single integrated broadband communications network is necessarily the ideal solution for all types of services.

There is thus a need for a system that can exploit the use of a shared physical access network for a range of different service and traffic types, whether they are more suitably supported by connection-less or connection-oriented transfer modes. An important additional need, is to optimise the bandwidth utilisation in existing access and core networks, while at the same time meeting, in an optimal manner, a user's requirements. These requirements include QoS, comprising the parameters for cell loss and delay statistics, and also issues such as security, cost, and availability.

Architectural Considerations

The architectural framework will consist of service-to-resource mapping (SRM) functionality, hosted in a user's residential gateway and one or more local access points, each of which in turn, communicates with a number of network operators and service providers. As figure 1 shows, a key feature of the SRM system will be the use of intelligent agent technology.

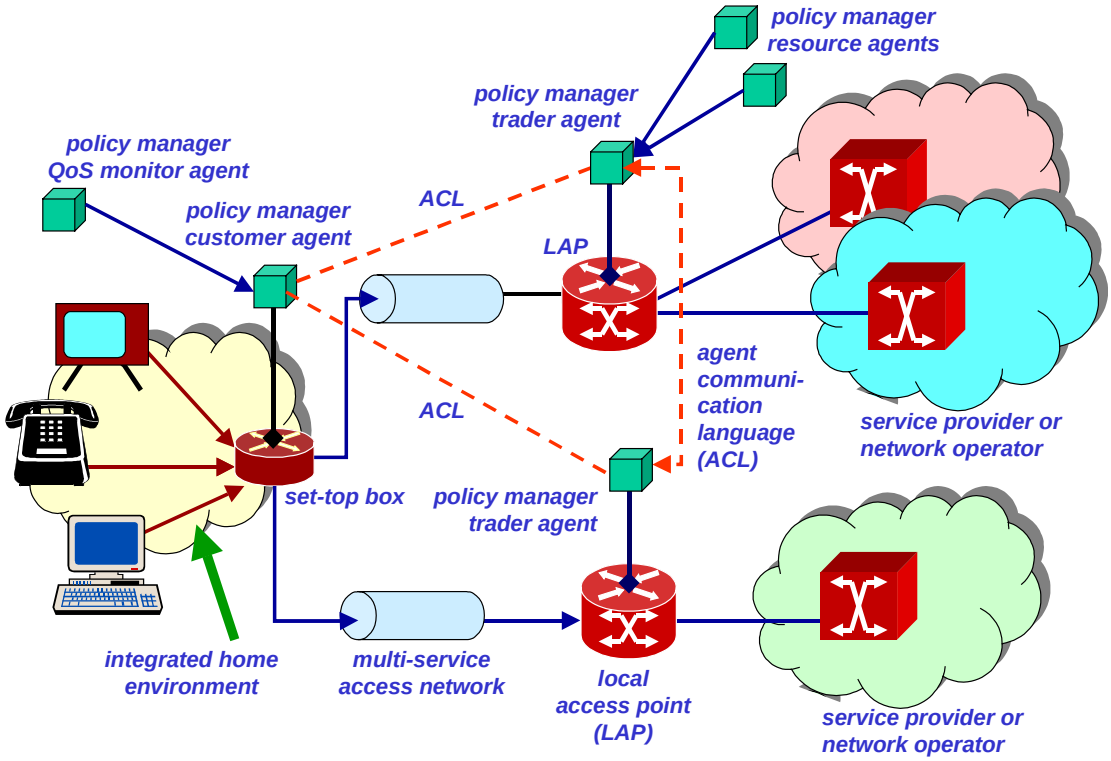


Figure 1: Overview of the Multi-Service Test-bed

The Home Network

Work in this area recognises that a major growth area will be the networking of, and inter-working between residential equipment such as telephones, PCs, televisions, consumer equipment for heating, lighting and security as well as equipment associated with the supply of heat, light and power.

A common protocol set will be developed to cover the areas of the customer access point, the customer terminal (set-top-box) and the home distribution system itself. Such a common protocol set will allow manufacturers of domestic equipment to install appropriate hardware and software interfaces to their products. Such interfaces will enable these products to be interconnected and controlled in a way that minimises the number of interfaces required.

The local loop provides the interconnection between domestic users and the local access points. A wide variety of local-loop technologies, based on copper, coax, fibre and radio, are likely to coexist for some time. Copper-based ADSL is presently a strong contender for the support of multi-media services in the local loop. However, whichever technology is used, provision must be made for a baseline telephone service for emergency situations, should a power failure occur.

The home network itself may be built on technologies based on Ethernet, mains power lines or radio (e.g. HIPERLAN or Bluetooth).

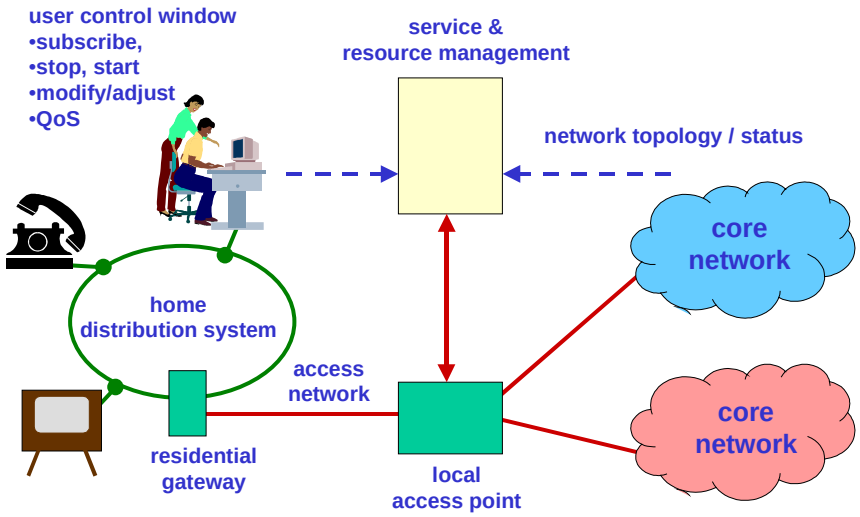


Figure 2: The Home Network

Local Access Point

The Local Access Point (LAP) can be regarded as a high-technology local exchange. It will provide customer negotiation facilities and also host accounting and security functionality. It will be able to handle authorisation of accesses to the customer for tasks such as metering, security monitoring and activation of residential equipment and devices. The LAP will also have facilities for Edge-of-Network-Processing (EoNP) to support consumer applications such as Video-on-Demand.

The LAP will have interfaces for a number of local-loop technologies and will connect to many local residential customers. These in turn, may each be connected to more than one LAP. On the core network side, the LAP will enable access to service providers and core networks, be they based on IP, ATM, SDH, ISDN or even POTS.

Each LAP will be made up from computer-controlled switching fabrics. The computers will host the most important parts of the service-to-resource-mapping system.

Service to Resource Management

The main application of the TORRENT test-bed is the implementation of a number of different architectures to support serve-to-resource management. Functionality will include the service negotiation, configuration and creation, control and re-negotiation. Re-negotiation may be in real-time. Typical multi-media services could be multi-media conferences or access to multi-media databases. The latter can comprise a number of phases, including browsing, download or replay in real time. Each phase will have its associated QoS, accounting and security requirements.

TORRENT will use agent technology for the SRM system. Agent technology reduces the need for centralised control and scales well with the size and capabilities of a communications network. It has been successfully applied in a number of communication projects dealing with the control of, and service provision in, ATM and mobile networks [1]. A software agent is a software entity that can act in an autonomous manner, can learn (be reactive) and be proactive. It can also interact with other agents, software systems and humans.

Customer agents will represent the interests of individual customers. Trader Agents will represent the service offerings of network operators and service providers. Agents will also represent accounting functionality as well as security and authentication processes. Importantly, agents will be able to negotiate on behalf of customers and service providers.

The design of the agent architecture will take cognisance of the IETF architecture [2]. This specifies three horizontal planes, namely the Policy Management, Control and User planes. The Policy Management plane involves Service Level Agreements, Policy Data Repositories relating customers to their preferred service options, as well as Policy Repository Server Agents, Customer, Monitor, Trader and Policy Consumer Agents. These will rely on control plane protocols to access User Plane Resources.

The extended markup language (XML) [3] is a candidate for inter-agent communication. XML allows information to be stored in structured files and ported across different platforms. It covers both the vocabulary and the syntax of communicated messages.

A FIPA-compliant [4] agent platform will host the SRM software. Such a platform facilitates the creation and deletion of agents, provides agent communication channels and also makes available a directory facilitator system for finding agents.

Field Trials

The field trials will establish scenarios involving distributed intelligent agent-based management and control procedures, in the application, home network, residential gateway, local access point and the core network. The objective will be to show that application traffic will be routed over the most appropriate access network (when a

choice is available) and the most appropriate core network to suit the instantaneous QoS requirement (mainly delay and loss statistics) vs. cost.

Scenario options include access technologies based on copper pairs and cable TV, power-line connectivity, local radio loop and optical fibre. xDSL and IP-over-ATM will feature in this range of options. Trials will not just be laboratory-based, but also envisage trans-European connectivity.

Field Trials will show that the TORRENT test bed can provide answers to the following performance issues, when new services or home access facilities are planned or introduced. Performance from the Viewpoint of the User or Customer includes issues such as ease of use, robustness, flexibility, efficiency, transparency, security, negotiation speed, set-up and release time, QoS (loss and delay statistics), cost effectiveness and ease of monitoring for charging and accounting. Performance from the Viewpoint of the network and service providers is concerned with the ability and effectiveness of the access and core technologies to support required combination of services and service components. These issues include ease of monitoring for performance, charging and accounting, and use of resources in the context of bandwidths, database needs negotiation and computation.

Conclusions

TORRENT will demonstrate the advantages for the customer of intelligent control over the choice of network operator and provider, and, parameters such as QoS, cost and security. This is relevant for the provision to the customer of multi-media services. The results of the project will have a strong impact on telecommunications standardisation.

References

[1] Bigham J. et al, Resource Management and Charging using a Multi-Agent System, IFIP, Queen Mary, University of London, April 1999

[2] The Internet Engineering Task Force, http://www.ietf.org/.

[3] Extensible Markup Language (XML), http://www.w3.org/XML/

[4] The Foundation for Intelligent Agents, Geneva, Switzerland, http://www.fipa.org.

Acknowledgements

While Queen Mary, University of London is leading the project, it should be emphasised that there are twelve partners in this consortium. Each partner has an important role. In addition to Queen Mary, the partners are, Portugal Telecom, Essto-Helleninc PTT Consulting, Telenor, Tesion, Flextel, Intracom, Versaware, MultiComLab, University of Stuttgart, Waterford Institute of Technology and Genuity Incorporated.

AN ACTIVE MODEL FOR FACIAL FEATURE TRACKING

Jörgen Ahlberg

Image Coding Group, Dept. of Electrical Engineering,
Linköping University, SE-583 31 Linköping, SWEDEN
Tel: +46 13 282163; fax: +46 13 284422
e-mail: ahlberg@isy.liu.se

ABSTRACT

We present a system for finding and tracking a face and extract global and local animation parameters from a video sequence. The system uses an initial colour processing step for finding a rough estimate of the position, size, and in-plane rotation of the face, followed by a refinement step driven by an Active Model. The latter step refines the previous estimate, and also extracts local animation parameters. The system is able to track the face and some facial features in near real-time, and can compress the result to a bitstream compliant to MPEG-4 Face & Body Animation.

1. INTRODUCTION

The goal of this work is to extract parameters describing the adaptation of a face model to the frames of a video sequence. Since the target applications are video-phones and man-machine interaction, we assume a "video phone-like" input, that is, we assume that there is a face in the image, looking approximately into the camera. We use the face model CANDIDE-3 [1], and the adaptation parameters are global (rotation, translation, scale) as well as local (Action Units [1][2] controlling the mouth and the eyebrows).

Initially, a colour based algorithm (described in Section 2), assuming that skin colour is recognizable, is used to give a rough estimate of the size and position of the face. Those parameters are then refined by an Active Model (described in Section 3). The two algorithms are presented in the subsequent sections, followed by the results, and a description of our ongoing work.

2. THE INITIAL STEP: THE COLOUR BASED ALGORITHM

To quickly find the approximate location of the face in the input image, the pixels are traversed and each given a likelihood value of been a skin colour pixel. This likelihood value is based on à priori collected statistics, to which a mixture of Gaussian distributions has been adapted using the Expectation-Maximization (EM) algorithm. The resulting image of likelihood values is blurred and then thresholded, and of the remaining objects in the image, the largest one is selected as the most probable face candidate. The position, size, and orientation of this "blob" is used as the initial estimate handed over to the refinement step. Examples of resulting estimates are shown in Figure 1.

This kind of algorithm has been chosen because it is fast and simple. The obvious drawback is that it need re-calibration for each camera.

3. THE ACTIVE MODEL

Our Active Model is a simplification of the Active Appearance Models invented by Cootes et al. [4][4], and we will here describe the model and how it is parameterized.

First, the CANDIDE model has been adapted to a set of images using 12 parameters: 3D-rotation, 2D-translation, scale, and 6 Action Units controlling the lips and eyebrows. We collect those parameters in a 12-D vector **p**, which thus parameterizes the geometry of the model.

The image under the wireframe model has, for each image in the training set, been mapped to the model, and the model has then been normalized to a standard shape, size, and position, in order to collect a geometrically normalized set of textures. On this set of textures, a PCA has been performed and the eigentextures (*geometrically normalized eigenfaces* [5]) have been computed.

We can now describe the complete appearance of the model by the geometry parameters **p** and an N-dimensional texture parameter vector, where N is the number of eigentextures we want to use for synthesizing the model texture. Given an input image and a **p**, the texture parameters are given by projecting the normalized input image on the eigentextures, and thus, **p** is the only necessary parameters in our case.

We can compare this to the parameterization used by the Appearance Models. For the Appearance Models, a PCA is performed to find the suitable subspace of *appearance modes* combining deformation modes and texture modes (eigentextures). In our application, we only parameterize the model in terms of deformation (including global motion) since we know in advance what kind of parameters we are interested in extracting. If we want to extract Action Units

Figure 1. Examples of the colour-based algorithm giving a rough initial estimate of the location, size, and in-plane rotation of the face.

(the parameters typically used for CANDIDE) we simply parameterize and train our model on those parameters (or deformations spanning the same subspace).

3.1. Active Model Training

It is possible to compute, for each set of parameters controlling the wireframe model, a match between an input image under the wireframe model and the reconstructed image using those eigentextures. We could then try all possible values of the model parameters, and regard the best ones as the optimal adaptation of the model to the input image. This would however be too time consuming for most applications.

One solution is to use the Active Appearance algorithm [3] used for adapting Appearance Models to images. As mentioned above, our model is not an Appearance Model, but even so we can use the principle of the Active Appearance algorithm.

The training procedure is as follows: For each of the training images, we change, one by one, the model parameters slightly and map the image to the model. We then try to reconstruct the new texture with our eigentextures, and compute the residual $\mathbf{r}$ between the reconstructed texture $\mathbf{x}$ and the mapped texture $\mathbf{j}$. We choose $e = \|\mathbf{r}\|^2$ as the error measure, and try to optimize it over the model parameter vector $\mathbf{p}$. The procedure is illustrated in Figure 2. Analysis of the residual image will give us information on how to update the parameter vector, as explained below.

If we Taylor-expand $\mathbf{r}$ around $\mathbf{p} + \Delta\mathbf{p}$ we get

$$\mathbf{r}(\mathbf{p} + \Delta\mathbf{p}) = \mathbf{r}(\mathbf{p}) + \mathbf{R}\Delta\mathbf{p} + O(\Delta\mathbf{p}), \quad (1)$$

where

$$\frac{\partial}{\partial \mathbf{p}}\mathbf{r}(\mathbf{p}) = \mathbf{R} \quad (2)$$

We approximate $\mathbf{r}(\mathbf{p})$ with the first terms and regard it as a linear function.

Given an initially suggested $\mathbf{p}$, we want to minimize

$$e(\mathbf{p} + \Delta\mathbf{p}) = \|\mathbf{r}(\mathbf{p}) + \mathbf{R}\Delta\mathbf{p}\|^2 \quad (3)$$

of which the least square solution is

$$\Delta\mathbf{p} = \mathbf{U}\mathbf{r}(\mathbf{p}) = -(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}\mathbf{r}(\mathbf{p}). \quad (4)$$

From a set of training data (models adapted to images) we can estimate $\mathbf{R}$, and from the estimated $\mathbf{R}$ compute the *update matrix* $\mathbf{U}$.

3.2. Active Model Search

Assuming a rough estimate of the model parameters (in this case given by the colour based algorithm), we can reconstruct the texture using the eigentextures, compute the residual image $\mathbf{r}(\mathbf{p})$ and use the à priori computed update matrix $\mathbf{U}$ to find out how we should change the parameter vector $\mathbf{p}$ to get a better model adaptation. That is, we update according to

$$\mathbf{p}' = \mathbf{p} + \mathbf{U}\mathbf{r} \quad (5)$$

and compute the new error measure

$$e' = \|\mathbf{r}(\mathbf{p}')\|^2. \quad (6)$$

If this is not an improvement, we try smaller update steps (0.5, 0.25). If there is still no improvement, we declare convergence. If the maximum absolute value of $\Delta\mathbf{p} = \mathbf{U}\mathbf{r}$ is smaller than a certain threshold (that should be smaller than an easily perceptible change in the model geometry), we declare convergence without computing the new error measure.

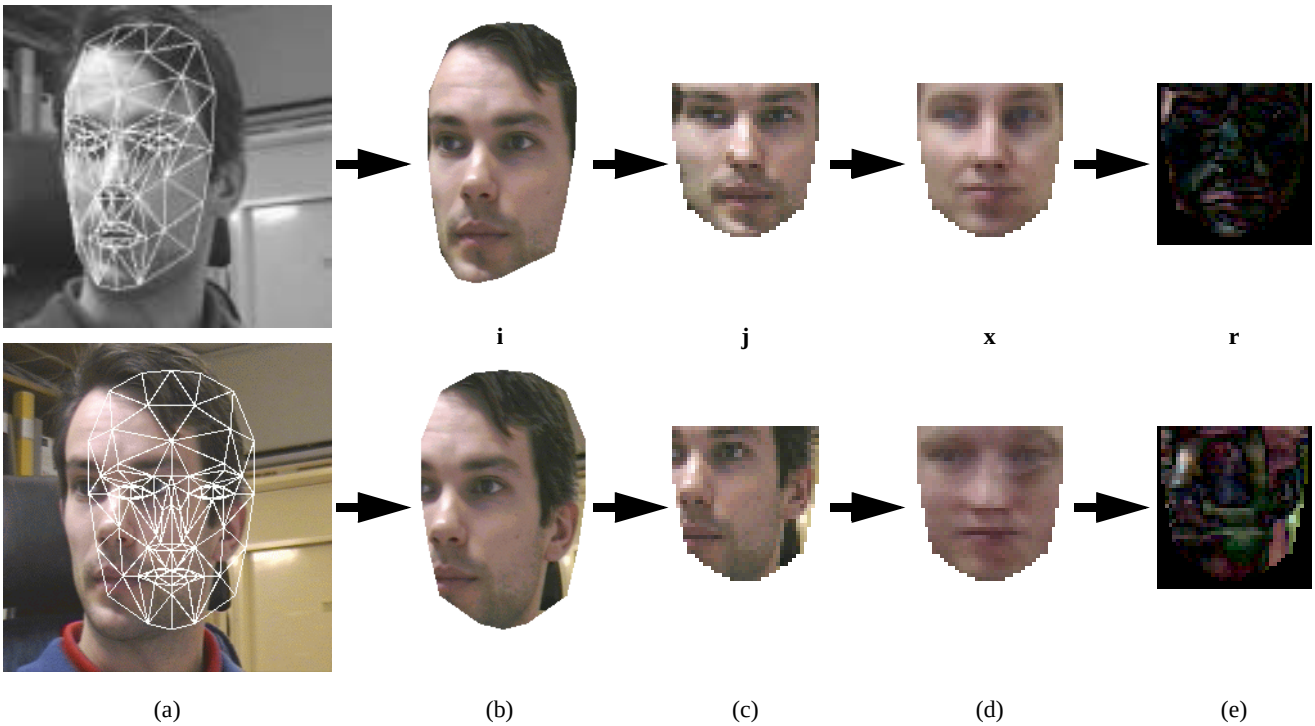


Figure 2. The model matching and texture approximation process. A good and a bad (top and bottom row respectively) model adaptation is shown (a); The image mapped onto the model (b); The model is reshaped to the standard shape, producing the image **j** (c); The normalized texture is approximated by the eigentextures, producing the image **x** (d); The residual image **r** is computed (e). The images **j** and **x** are more similar the better the model adaptation is. Analysis of the image **r** tells us how to improve the model adaptation, that is, how to minimize the difference between **j** and **x.**

For the following frame, the initial estimate is given by the refined estimate in the previous frame, and the colour based algorithm is thus used for the first frame only.

4. IMPLEMENTATION

We have implemented a C++ library with routines for handling face models and training them as to be Active Models. The shape of the (normalized) eigentextures is determined by the standard shape of the CANDIDE model (with the upper part of the head removed) scaled so that the size is 40x42 pixels (see Figure 2 c-d). The eigentextures have been computed in RGB as well as in grayscale for comparison, as have the update matrix **U**.

The implementation uses OpenGL for the texture mapping in the Active Model Search, utilizing the fact that modern graphics cards have specialized hardware for such tasks. The geometrical normalization of the input image (see Figure 2 c) is thus performed in a very short time (less than 2 ms), and the speed of the algorithm is dependent more on the graphics card than of the CPU.

4.1. MPEG-4 encoding

The animation of the face model can be encoded using the MPEG-4 standard for face animation. Since the vertices of the CANDIDE-3 model corresponds quite well to the facial feature points defined in MPEG-4, the coding is easily done.

The Facial Animation Parameters (FAPs) used to represent movements of the facial feature points in MPEG-4 are measured in face dependent scales, using different FAP Units (FAPUs). The FAPs are also measured relative to the neutral face, and thus a neutral face model is kept in memory. Using this neutral face model, the FAPUs and the FAPs are computed, and then compressed using the MPEG-4 reference software. The entire process takes only a few milliseconds and does not influence the real-time performance. The output is an MPEG-4 Face & Body Animation (FBA) compliant bitstream, that can be played in an FBA player, for example FAE [6]. The bitstream can be stored on a file or streamed over the network.

5. RESULTS

The experiments presented here are performed on a PC with a 500 MHz Intel Pentium III processor and an ASUS V3800 graphics card with video input. The colour based

algorithm runs on approximately 0.1 seconds, and the Active Model search needs about 15 ms per iteration. Typically, less than 10 iterations are needed each frame, and fewer iterations are needed the closer the initial estimate is to the optimum. Thus, if a video sequence is recorded at a high framerate (with small motion between each frame), the tracking will also run on a higher speed. Visual results are shown in Figure 3.

Using grayscale eigentextures and update data, it turned out that the computation in the graphics card (which internally uses RGB) became almost 20% slower. However, the computations performed in the CPU became (as expected) about 3 times faster, and then only 20% of the total computing time is due to the CPU (the rest being computations in the graphics card).

Testing on a video sequence of a few hundred frames gave results according to Table 1 below. It is clear that the grayscale computations are preferable, since the visual results are equivalent.

Table 1: Timing results (average over 341 frames)

Measurement	RGB	Grayscale
Iterations per frame	6.9	6.8
Total time per frame (ms)	94.1	69.1
Time for computing $\Delta\mathbf{p}$ (ms)	7.2	5.05

6. CURRENT WORK AND FUTURE IMPROVEMENTS

There are several ways this system can be improved, and they are currently under investigation. Four things to be considered are mentioned here:

First, the colour based algorithm is not robust enough, and it should be complemented with some more simple & fast technique. For example, we could require that an area could be regarded as a face only if there is with some difference (due to motion) between the first and second frame.

Second, **U** is a somewhat sparse matrix. By utilizing this fact, the computation time could be improved.

Third, as all tracking systems, this system can lose track, and therefore, some kind of re-initialization scheme is needed. On possible procedure is that when the Active Model does not converge to a small error measure, the colour-based algorithm is invoked, handing a new initial estimate to the Active Model.

Fourth, our currently used set of Action Units is not complete. For example, we do not analyse the motion of the eyelids at all, and the shape of the head is assumed to be known à priori.

7. CONCLUSION

We have presented a system that tracks a face and facial features in a video sequence. The resulting animation data is encoded using MPEG-4 Face Animation. The system works in near real-time, and the experimental results are promising. With some further development and optimization, a real-time 3D face and facial feature tracker should be possible to implement on consumer hardware.

8. ACKNOWLEDGEMENTS

This work is financed by the European IST-project InterFace and the national Swedish SSF-project VISIT. The author is also thankful to Dr. Tim Cootes for advice on the Active Appearance algorithm.

9. REFERENCES

[1] J. Ahlberg, *CANDIDE-3 - an updated parameterized face*, Report No. LiTH-ISY-R-2326, Dept. of EE, Linköping University, Sweden, January 2001.

[2] P. Ekman and W. V. Friesen, *Facial Action Coding System*, Consulting Psychologist Press, 1977.

[3] T. F. Cootes and C. J. Taylor, *Statistical Models of Appearance for Computer Vision*, Draft report, Wolfson Image Analysis Unit, University of Manchester, http://www.wiau.man.ac.uk

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," *Proc. 5th European Conference on Computer Vision*, pp. 484 - 498, 1998.

[5] J. Ström et al., "Very Low Bit Rate Facial Texture Coding," *Proc. IWSNHC3DI*, Rhodes, Greece, September 1997, pp. 237 - 240.

[6] F. Lavagetto, R. Pockaj, "The Facial Animation Engine: towards a high-level interface for the design of MPEG-4 compliant animated faces," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 2, March 1999. Software available at http://www-dsp.com.dist.unige.it/~pok/RESEARCH/MPEG/fae.htm

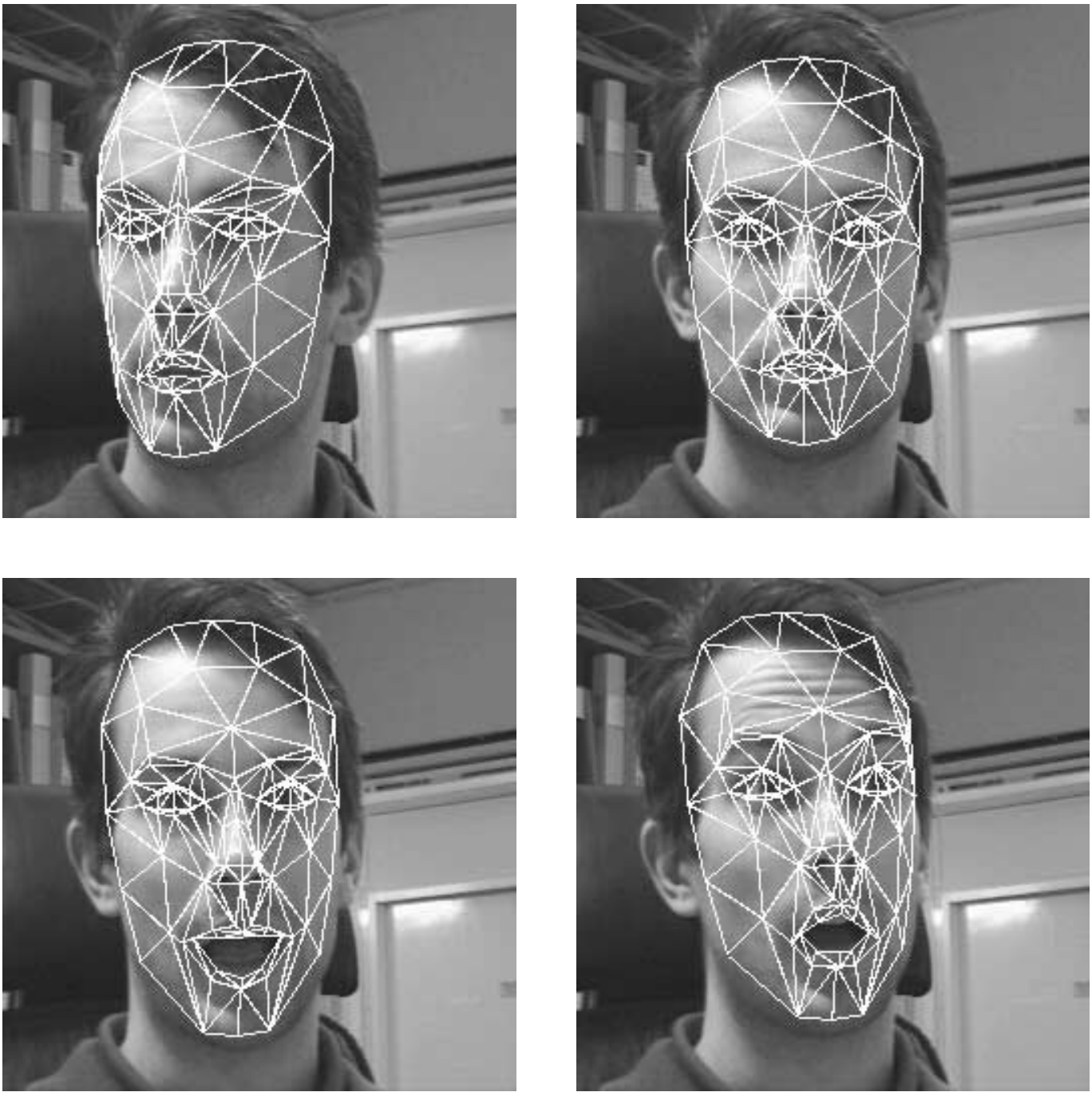


Figure 3. The CANDIDE-3 model adapted to four frames of a video sequence.

SIMILARITY MEASURES FOR NATURAL LANGUAGE IMAGES

Julia A. Johnson
Department of Math & Computer Science, Laurentian University
Sudbury, Ontario P3E 2C6, Canada, E-mail: julia@cs.laurentian.ca

Tracy C. Mansfield
Conceptual Imagery™ Project, International Neural Machines
Waterloo, Ontario N2T 2K9, Canada, E-mail: tmansfie@ineural.com

ABSTRACT

Natural languages such as English, French, German, sign language (for example ASL) or Chinese are inextricably linked with images. Humans articulate complicated sentences based on images that they conceptualize. We have previously proposed an imagistic model of language. Our interest here is to apply current image processing techniques, specifically similarity measures, to natural language images. Rough sets and rough mereology provide the theoretical basis for our proposal. This application is of interest for a computational implementation of imagistic rather than symbolic processing.

1 INTRODUCTION

Problem statement is provided in Section 2 along with an example to illustrate the concept of a natural language (NL) image. Existing solutions and their criticisms are provided in Section 3. In Section 4 we demonstrate how similarity measures and quality information can be used to process NL images. In particular, rough sets and rough mereology are applied to the problem of natural language images. Potential advantages of image processing of NL images are discussed in Section 4. Conclusions are presented in Section 5.

2 OUR PROBLEM

Our objective is to find a way to describe appropriate image query/target image pairings. This is achieved by providing the computer with many examples of good pairings. The computer recognizes a pattern and writes a specification for what a good query to target pairing would be.

Table 1 provides a list of descriptors for images. We wish to develop rules based on image descriptors that will help to predict whether or not an image pairing is appropriate. Those predictive rules are induced from the example images IMG1, IMG2 and IMG3 described in the information table.

Graphics	IMG1	IMG2	IMG3
Centriod	acentric	centric	acentric
Curvature	straight	curved	straight
Shape	round	curved	flat
Size	small	medium	large
Sound	poetic	factual	factual
Cohesion	disjoint	cursive	disjoint
Structure	l-r	t-d	l-m-r
Acceptable?	yes	no	yes

Table 1: Criteria for Describing Images

Definitions from [9] for image properties are as follows: 1) *Centriod*: geometric appearance of a character computed from its pixel patterns. 2) *Curvature*: whether the image consists mostly of straight lines or curved lines, detected from the distributions of pixel patterns. 3) *Shape*: possible values $\{round, square, flat\}$. 4) *Size*: possible values $\{small, medium, large\}$ referring to, for example,the number of strokes comprising a Chinese character. 5) *Sound*: possible values $\{poetic, factual\}$ expressing something of the meaning of the image. 6) *Cohesion*: possible values $\{disjoint, cursive\}$ indicates the extremes between parts of images weakly joined or strongly joined (e.g., by lines). 7) *Structure*: possible values $\{left-right, top-down, left-middle-right, inseparable\}$ applicable to Chinese characters which have these four kinds of structure abbreviated l-r, t-d, l-m-r, in, respectively.

In this paper, we will illustrate how uncertainty reasoning techniques can be used to automatically develop rules for querying based on images.

3 EXISTING SOLUTIONS

We are not aware of previous work in which image processing technigues are applied to imagistic representations of language. Mansfield [13] and Kohanim & Johnson [10, 1] research the subject of imagistic representation of language. These works are reviewed here and a common terminological framework is developed to facilitate our discussion.

The first chapter of Mansfield [13] details the association of various stages of language with different degrees of being internal or external, including a good definition of what defines that boundary. Any part of language which is pre-sensory relative to agent L, or post-articulatory relative to agent S, is "external," and *everything* else is "internal." The reason that a distinction was made between perception, cognition, and articulation was precisely to distinguish *among* internal states, for a wide variety of reasons including the need for well-defined terms when talking about how humans make a kinaesthetic appeal to their experience with language during perception. In any case, the *only* part of language which is external is the signal itself. Therefore, *both* images *and* concepts are internal representations of such signals.

The trick is to consider a *symbol* as bipolar: A symbol is the association of a concept/image with a conventionalized articulatory form. Some such forms are highly *iconic*, such as the word "woof", and some of them are highly *arbitrary*, such as the form "tree." Notice that even a concept which is conventionally held to be quite concrete, like "tree", can still have a wholly arbitrary form – there is no necessary relation between concreteness/abstractness and iconicity/arbitrariness. Having said that, the term "image" *tends* to be used *as if* images represented conceptual structures which were more concrete, and that the term "concept" *tends* to be used not only *as if* it were a more generic term, but also to cover more abstract and more complex conceptual structures. More generally, semantic structures are those conceptual structures that are evoked by linguistic expressions. Humans appeal to all sorts of conceptual structures, but only the ones evoked by linguistic signals are considered semantic. When a word is perceived as only so much noise, such as when it is uttered in a foreign language, then it is not linguistic. It might still be *communicative*, but not linguistic, hence not semantic. These matters are covered at length in Mansfield [13].

Quoting Kohanim & Johnson [10], entities, states, context, and images are used to build a state-grammar and image-reasoning framework within which both natural language syntax and semantics emerge as products of an agent's interaction with its external and internal worlds. A Concept corresponds to the intension (Note: "intension" implies "sense", which is internal) of a set and an Image to the extension (Note: "extension" implies "reference", which is external). The extension changes with time and hence there are different Image instances for a given Concept at different instances in time.

Johnson & Kohanim [1] have depicted language to be a sub-domain of the external world termed the World of Manifests. All the entities in the external world are included in the sub-domain of real world primitives 'RWPs'. An agent is an RWP. Since we live in a time/space continuum, the RWPs and Manifests may have different characteristics in time/space continuum. In this respect, the mapping between RWPs and Manifests is a relationship (not a function) defined in a time/space continuum. A Manifest could reference many RWPs (and their information/attributes/events/states). For example, 'class' could be a course or a social class. An RWP can reference many Manifests. Since manifests are part of the external world, they could reference other manifests as if they were an RWP that was referencing a Manifest or vice versa. It is the nature of this relationship that causes natural language ambiguities (there is no one-to-one mapping between RWPs and Manifests).

Agents utilize their sensory devices to interact with their surroundings. They utilize the same sensory devices (+ motor capabilities) specifically sight, sound, and touch to interact with a language construct. Each sensory device produces a different type of image for the same Manifest. Images are non-discrete entities with basic information about the external entity that is being referenced. A categorization of modalities is assumed in [11, 10, 1]. In contrast, Table 2 crosses modality with discourse component which we think is a superior classification to that which has appeared before.

	Discourse Component	
Modality	Transmission	Sensation
Sound :	Speaking	Hearing
Light :	Signing	Seeing
Touch :	Braille	Feeling

Table 2: Modality vs. Discourse Component

Ancillary modes could be added where speaking can also appeal to light, and signing can also appeal to sound or touch.

The intent in Mansfield [13] is in part to show that the course of the development from sensation to language (through communication) involves an increase in the ability to handle images of a more ab-

stract nature, where classification and structuring of these images is part of that abstraction. Those more sophisticated images still have an iconic base, and any process of abstraction away from that base is still going to be motivated, rather than arbitrary. In other words, we should never have a conceptual structure that is just a red ball, for example, tied to concepts of a blue square. In fact, in work by the team of George Lakoff and Mark Johnson (studies of metaphor) they try to spell out the ways in which this structuring is motivated. Work by Langacker shows motivated cognitive 'operations' on these images, and the reader is referred there for a discussion of what image reasoning might consist of. In [13], the evolution of two primitive functions from sensation to language is posited.

Given a sample query image and a set of target images, to avoid translation of images to text, we propose an application of existing image analysis techniques to get the query image to match up with the correct target.

In contrast, CI™ (from International Neural Machines) submits images to data mining techniques in lieu of a text structure (similar to the way in which INM already mines medical images). However, for the older CM™ and the newer CI™ technologies the structure of images cannot be described without compromising the patent submission.

4 SUGGESTED SOLUTION

The training set should be large to ensure the generation of rules that are representative of the image space. At a cost of describing, by means of image attributes, a large number of images for the training set, we are able to provide the mapping of query images to targets in a set of rules applicable to new (query image, target image) pairs (i.e., pairs that do not initially appear in the training set). This is an advantage because rather than requiring all images in the space to be translated to text, only a representative sample must be described in terms of their properties.

4.1 Rough Sets

The notion of a *rough set* was introduced by Zdzislaw Pawlak [14] to deal with incomplete and inconsistent domains. A variety of applications of rough sets have been undertaken including software specification [6], deadlock detection in Petri Nets [7], database query [3, 2], education in the www [12], scheduling [5] and natural language processing [9, 11, 10, 1]. Some definitions of basic concepts follow:

An *indiscernibility class* with respect to set of attributes A is defined [15] as a set of examples all of whose values for attributes $a \in A$ agree. For example, the indiscernibility classes with respect to attributes $A = \{Centroid, Curvature\}$ of Table 1 are $\{\text{IMG2}\}$ and $\{\text{IMG1}, \text{IMG3}\}$.

Not all of the attributes are necessary for prediction. If a set of attributes and its superset define the same indiscernibility classes, then any attribute that belongs to the superset but not the subset is redundant. An information table with no redundant attributes is said to be minimal.

Let X be a concept (considered as a subset of the entities in the domain) that we wish to approximate. A rough set approximates X by a pair of sets $\underline{X}$ and $\overline{X}$ which give lower and upper approximations to X.

4.2 Quality

We wish to predict characteristics of good image pairings based on the example images represented by the minimal information table. One such rule is 'if the image is acentric and the curvature is straight and the cohesion is disjoint then the image is acceptable'.

A possible measure of the strength of a rule is, for example [15]:

$$\frac{\#\ of\ positive\ examples\ covered\ by\ the\ rule}{\#\ of\ positive\ and\ negative\ examples\ covered\ by\ the\ rule}$$

By this definition the rule (*Centriod*, acentric) and (*Curvature*, straight) $\longrightarrow$ (acceptable, yes) has a strength (or probability) of $\frac{2}{3}$. In general, when there is a choice of applicable rules which differ in their probabilities we choose to apply the higher probability rule.

Rough set theory helps us to formulate decision rules when the information table is inconsistent. At the moment, every minimal table that can be obtained from Table 1 is consistent. However, consider the addition of an image to the table with values for the attributes respectively acentric, straight,*,*,*,*,* where '*' stands for a wild card. Assume that the decision attribute for this new entry is 'no'. This table is inconsistent because the following rule induced from the table is inconsistent: 'if the image is acentric and the curvature is straight then the system is *both* acceptable and unacceptable'.

4.3 Similarity

This exposition parallels a previous presentation in which rough mereology (RM) is explored for its applications to software specification [6]. Here we are interested in RM application to image query processing. Similarity measures based on a set of examples is a superior approach for image query processing because translation of the image to text is not required. Rules are automatically generated based on examples of what are considered satisfactory relationships between image queries and target images.

Rough mereology is the theory of the "part of to a degree" relation developed by Polkowski and Skowron [16]. Their rough inclusion function $\mu(x, y)$ is read *the degree to which x is a part of y*.

For simplicity, let us measure similarity between images by the proportion of attribute values that the images have in common. See also similarity measures provided in [4, 8]. We now illustrate computation of $\mu(x, y)$ from Table 1. Assume a minimal table with only the attributes *Centriod*, *Curvature* and *Cohesion*.

System	IMG1	IMG2	IMG3
IMG1	1	0	.5
IMG2	0	1	.17
IMG3	.5	.17	1

Table 3: Rough Inclusion for Table 1

Table 3 gives ε values regarding the systems described in Table 1. A system shares all attribute values in common with itself (hence, 1's along the diagonal of Table 3). IMG1 shares no attribute values in common with IMG2 and 50% of attribute values in common with IMG3. The proportion of attribute values that IMG2 and IMG3 have in common is $\frac{1}{6}$.

Cost	IMG1	IMG2	IMG3
Compression	low	moderate	high
Storage	low	high	high
Transmission	high	low	low
Replication	high	high	low
Fragmentation	little	great	great
Acceptable?	yes	yes	no

Table 4: Similarity Criteria Dictated by Cost

Table 4 shows cost criteria which may show useful for gauging the similarity between images, specifically the cost to compress, store, transmit, replicate and fragment images.

Tradeoffs within the table are evident. For example, cost of compressing images reduces storage, transmission and replication costs suggesting that not all of the attributes are required to classify images (some are redundant and could be eliminated).

Interest in images from yet another perspective is illustrated by Table 6 which shows image processing and analysis operations. This choice of operations and their definitions have been adapted from the operations supported by image capture and processing software IMAGE SAVANT Version 1.0 for Windows NT (http://ioindustries.com). 40 standard operations are

System	IMG1	IMG2	IMG3
IMG1	1	0	0
IMG2	0	1	.4
IMG3	0	.4	1

Table 5: Rough Inclusion for Table 4

supported classified as user interface features, bitwise arithmetic (and, or, xor, ...), graphics, arithmetic (difference, add, average, max, min, multipy), analysis & measurement, image transformation and geometric (mirror, rotate, point-to-point measures ...).

Only a few of the operations have been selected for illustration, specifically those classified as geometric and within that class those classified as point-to-point. Definitions for point-to-point image operations from IMAGE SAVANT web site follow: 1) *coordination extraction*: extract the image coordinates of point locations using a cursor in the image 2) *line segment length*: measure the line segment lengths between pairs of point locations 3) *path length*: calculate the total of all segment lengths measured between points in a series of points 4) *segment angle*: measure the angle between a line segment and a horizonal plane 5) *polygon area*: calculate the area within a polygon described by a series of three of more points 6) *pixel value extraction*: extract the pixel values at point locations within an image 7) *plotting operations*: generate scatter and line plots from a set of points 8) *pixel copy*: copy pixels from a region of interest to another location within the same image buffer or a different image buffer.

Operations	IMG1	IMG2	IMG3
Coord Extr	easy	difficult	difficult
Line Seg Len	easy	moderate	difficult
Path Len	easy	easy	difficult
Seg Angle	easy	easy	difficult
Polygon	easy	difficult	difficult
Pixel Extr	easy	moderate	difficult
Plotting	easy	moderate	difficult
Pixel Cp	easy	moderate	difficult
Acceptable?	yes	yes	no

Table 6: Operations Supported on Images

Abbreviating graphics, operations and cost, respectively, G, O, and C let us allocate our problem to a 2-level agent hierarchy in the following way: Subagents G and O provide rules for specifying graphics and operations criteria, respectively. The top level agent C provides information about the cost of im-

System	IMG1	IMG2	IMG3
IMG1	1	.25	0
IMG2	.25	1	.25
IMG3	0	.25	1

Table 7: Rough Inclusion for Table 6

ages. The information tables for agents G, O and C are, respectively, Tables 1, 6 and 4 whose rough inclusion functions are provided in Tables 3, 7 and 5, respectively.

We wish to learn a function $\mathcal{F}$ that provides a description of the cost of an image object from components of the image expressed in terms of topological (graphical) properties and operations supported. Let G_i, O_i and C_i, respectively, denote the graphics, operations and costs for a particular image i. Consider function $\mathcal{F}(\varepsilon_G, \varepsilon_O) \longrightarrow \varepsilon_C$. From the rough mereology, if $C_1 = \Lambda(G_1, O_1)$ and $C_2 = \Lambda(G_2, O_2)$ then $\mu_G(G_1, G_2) = \varepsilon_G$, $\mu_O(O_1, O_2) = \varepsilon_O$ and $\mu_C(C_1, C_2) = \varepsilon_C$ where $C_i = \Lambda(G_i, O_i)$ means the cost of image i derives from its graphical and operational properties. A standard algorithm RS1 for generating rules according to the rough sets paradigm implemented in Java [18, 17] permits the learning of function $\mathcal{F}$.

4.4 Natural Language Images

A framework for language processing based on images rather than symbols is proposed. We have a stimulus generator in the environment (which is what would normally be termed the 'referent', but which we leave unlabeled), and we have the internal collection of sensations/perceptions which the generated stimulus evokes. This collection is what we are calling an Image, or internal analog to the referent. Furthermore, we are portraying cognition as the mental manipulation of these images, rather than any more abstract symbols.

We have been talking about words being symbols, and symbols being relations between forms and meanings. While this notices the connection between the thought [DOG] and the encoded signal "dog", it misses the fact that there is an internal representation of the encoded signal. The kinaesthetic appeal to a person's own experience with generating that signal is explained in [13], but it must be stressed that the entity to which this appeal is made is the internal representation of the conventional form of that signal. There will naturally be variance in this representation from person to person, but we are able to communicate only because overlap exists.

4.5 The Image Lexicon

Consider a passive agent that does not know any words and has no capacity to memorize them (More on memorization later). This agent, for each object (real world primitive RWP) that it needs to reference, has to open up a dictionary (lexicon) and search through a set of tuples of the form:

$< picture\ of\ the\ referenced\ object >=$
$\{< picture\ of\ word_1 >, \ldots, < picture\ of\ word_n >\}$

Let us call this $Form\#1$. Since words have different (but not mutually exclusive) representations in an external world, the agent has to open up another dictionary depending on the desired modulation, conforming to the following tuples:

$< picture\ of\ word >=< type\ of\ modulation >$
$< instructions\ on\ how\ to\ modulate\ depending\ on$
$agent's\ internal\ structure\ for\ motor\ movements >$

Let us call this $Form\#2$. Each agent could have a different set of instructions that would produce the same word in a desired modulation (with some phonetic variations). For instance the instructions given to a Franklin dictionary (to utter a word) is different than the instructions given to an Intel machine running ViaVoice (IBM's voice recognition program) under Windows 95. Therefore, although the processor is internal to the agent, neither the instructions nor the words are internal to the agent. Instructions are agent dependent while lexicons are not.

Now consider the scenario in which the agent is capable of memorizing the (word $\longrightarrow$ instruction) mapping (for each type of modulation). In this scenario, the agent doesn't need to refer to dictionary #2 anymore since it is now internalized. However, the representation of this mapping (internally) is totally different than the one in the external dictionary (e.g., a set of neurons firing or bit streams).

Now consider the case in which the agent memorizes the (picture $\longrightarrow$ word) mapping. In this scenario, the agent does not need to refer to dictionary #1 anymore. However, again, the representation of this mapping is totally different than the one in the dictionary.

Here we have used only phonemes and morphemes since lexicon, discourse, prosody, and syntax (state grammar, school grammar) all utilize different variations of morphemes and phonemes (i.e., phonemes and morphemes are the basic building blocks for languages).

In conclusion, we have a lexicon that lives in parallel with our world. Each agent in our world is capable of internalizing only a subset of these lexicons. The instructions to reproduce or to map to RWPs

is agent dependent and the process to decipher this mapping in context is what we are referring to as Image Reasoning.

5 Conclusion

The novelty of our approach lies in the application of uncertain reasoning techniques to image processing with an extension to imagistic representation of language. We provide the computer with many examples of a query image appropriately matched with a target image. The computer recognizes a pattern and writes a formula for what an appropriate target image would be. But you cannot expect the computer to produce a good target image for a given query, based on just any examples of (query image, target image) pairings. The training set must be representative of "good" query image – target image matchings. This is possible because the rough set/rough mereology paradigm makes it facilitates imprecise solutions to problems rather than being restricted to precise answers.

The contribution in this paper has been to demonstrate the feasibility of applying approximate reasoning techniques to the problem of similarity measures for natural language imaging. We eliminate the need to translate such images into a text equivalent for subjection to other data mining techniques.

Acknowledgement

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERCC).

References

[1] J. Johnson and M. Kohanim. Learning based on communication and refinement. In *Proc. Workshop on Image Analysis for Multimedia Interactive Services WIAMIS'99*, pages 105–108, 1999.

[2] J. Johnson and M. Liu. The language Refine. In *Proc. 4^{th} Joint Conference in Information Science JCIS'98*, volume II, pages 367–370, 1998.

[3] J. Johnson and M. Liu. Rough sets for informative question answering. *Journal of Computing and Information*, 3(1), 1998. Available on-line http://www.jci.trentu.ca/jci/vol.3.

[4] J.A. Johnson. Semantic relatedness. *Computers & Math. with Applications*, 29(5):51–64, 1995.

[5] J.A. Johnson. Rough scheduling. In *Proc. 5^{th} Joint Conference in Information Science JCIS 2000*, volume 1, pages 162–165, 2000.

[6] J.A. Johnson. Specification from examples. In *Proc. International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, L'Aquila, Italy, July 31-August 6, 2000.

[7] J.A. Johnson and H. Li. Rough set approach for deadlock detection in Petri nets. In *Proc. International Conference on Artificial Intelligence IC-AI*, pages 1435–1439, 2000.

[8] J.A. Johnson and R.S. Rosenberg. A measure of semantic relatedness for resolving ambiguities in natural language database requests. *Data & Knowledge Engineering*, 7(3):201–225, 1992.

[9] J.A. Johnson, X. D. Yang, and Q. Hu. Word sense disambiguation in the rough. In *Proc. Fourth Symposium on Natural Language Processing SNLP*, pages 206–223, 2000.

[10] M. Kohanim and J. Johnson. Natural language imaging. In *Proc. Pacific Association for Computational Linguistics PACLING'99*, pages 114–130, 1999.

[11] M. Kohanim and J. Johnson. Using images as a foundation for natural language processing. *Computational Intelligence*, 16(4), 2000.

[12] A. Liang, M. Maguire, and J.A. Johnson. Rough set based Webct learning. In *Proc. First International Conference on Web-Age Information Management WIAM 2000*, pages 425–436, 2000.

[13] T.C. Mansfield. *Prominence: from Sensation to Language*. PhD thesis, UC–San Diego, 1997.

[14] Z. Pawlak. *Rough Sets-Theoretical Aspects of Reasoning about Data*. Kluwer, 1991.

[15] Z. Pawlak, J.W. Grzymala-Busse, Roman Slowinski, and Wojciech Ziarko. Rough sets. *Communications of the ACM*, 38(11):89–95, 1995.

[16] Lech Polkowski and Andrzej Skowron. Rough mereology: A new paradigm for approximate reasoning. *Int. Journal of Approximate Reasoning*, 15(4):333–365, 1996.

[17] R. H. Warren. Domain knowledge acquisition from solid waste management models using rough sets. Master's thesis, University of Regina, Regina, Canada, 2000.

[18] R. H. Warren and J. A. Johnson. A Java implementation of the RS1 algorithm using SQL. Technical Report TR-2000-03, University of Regina, Dept. of Computer Science, 2000.

OVERVIEW OF THE MPEG-7 STANDARD AND OF FUTURE CHALLENGES FOR VISUAL INFORMATION ANALYSIS

Philippe Salembier
Universitat Politècnica de Catalunya, SPAIN
e-mail: philippe@gps.tsc.upc.es

ABSTRACT

This paper presents an overview of the MPEG-7 standard: The Multimedia Content Description Interface. It focuses in particular on visual information description including low-level visual Descriptors and the Segment Description Schemes. The paper also discusses some challenges in visual information analysis that will have to be faced in the future to allow efficient MPEG-7-based applications.

1 INTRODUCTION

The goal of the MPEG-7 standard is to allow interoperable searching, indexing, filtering and access of audio-visual (AV) content by enabling interoperability among devices and applications that deal with AV content description. MPEG-7 specifies the description of features related to the AV content as well as information related to the management of AV content. As illustrated in Fig. 1, the scope of the standard is to define the representation of the description. For most of the description tools, the standard does not involve normative tools for the generation nor for the consumption of the description. However, as will be discussed in this paper, in order to guarantee interoperability for some low-level features, MPEG-7 also specifies part of the extraction process.

MPEG-7 descriptions take two possible forms: (1) a textual XML form suitable for editing, searching, and filtering, and (2) a binary form suitable for storage, transmission, and streaming delivery. Overall, the standard specifies four types of normative elements illustrated in Fig. 2: Descriptors, Description Schemes (DSs), a Description Definition Language (DDL), and coding schemes.

In order to describe AV content, a set of Descriptors has to be used. In MPEG-7, a *Descriptor* defines the syntax and the semantics of an elementary feature. A Descriptor can deal with low-level features, which represent the signal characteristics, such as color, texture, shape, motion, audio energy or audio spectrum as well as high-level features such as the title or the author. The main constraint on a descriptor is that it should describe an elementary feature. In MPEG-7, the syntax of Descriptors is defined by the *Description Definition Language (DDL)* which is an extension of the XML Schema language [5]. The DDL is used not only to define the syntax of MPEG-7 Descriptors but also to allow developers to declare new Descriptors that are related to specific needs of their application.

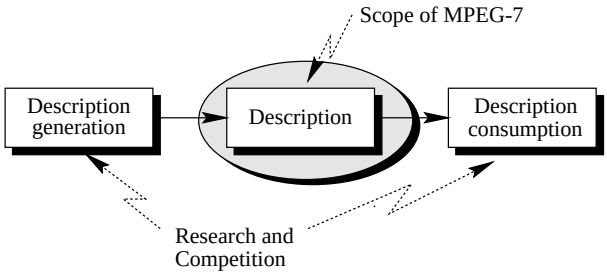


Figure 1: Scope of the MPEG-7 Standard

In general, the description of AV content involves a large number of Descriptors. The Descriptors are structured and related within a common framework based on *Description Schemes, (DSs)*. As shown in Fig. 2, the DSs define a model of the description using as building blocks the Descriptors. The syntax of DSs is also defined with the DDL and, for specific applications, new DSs can also be created.

When the set of DSs and Descriptors is instantiated to describe a piece of AV content, the resulting description takes the form on an XML document [4]. This is the first normative format in MPEG-7. This format is very efficient for editing, searching, filtering and processing. Moreover, a very large number of XML-aware tools are available. However, XML documents are verbose, difficult to stream and not resilient with respect to transmission errors. To solve this problem, MPEG-7 defines a binary format *(BiM: Binary format for Mpeg-7)* and the corresponding encoding and decoding tools. This second format is particularly efficient in terms of compression and streaming functionality. Note that XML and BiM representations are equivalent and can be encoded and decoded losslessly.

The objective of this paper is to provide an overview of the MPEG-7 DSs and Descriptors focusing on the visual aspects (section 2) and then, to discuss a set of visual information analysis challenges that could be studied to lead to very efficient MPEG-7-based applications (section 3).

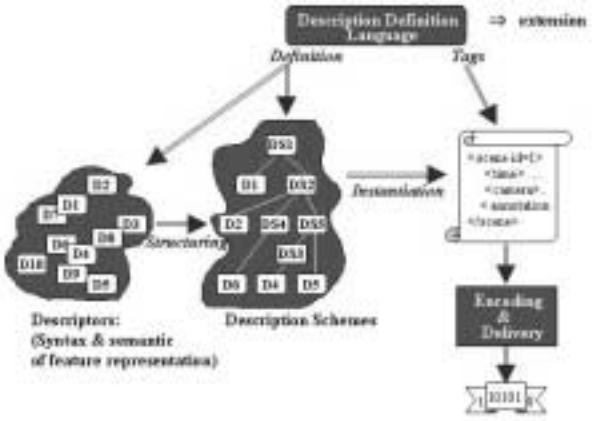


Figure 2: Main components of the MPEG-7 Standard

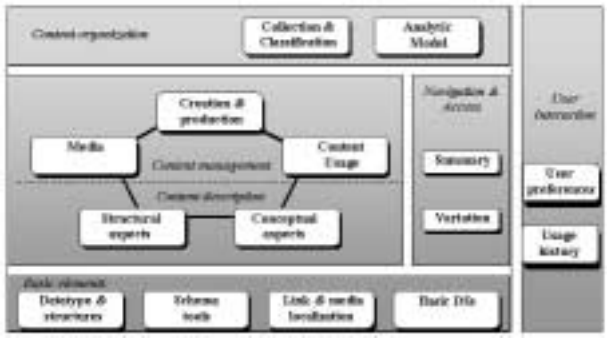


Figure 3: Overview of MPEG-7 Multimedia DSs

2 OVERVIEW OF MPEG-7

2.1 Multimedia Description Schemes

Fig. 3 provides an overview of the organization of the Multimedia DSs into different functional areas: Basic Elements, Content Management, Content Description, Navigation and Access, Content Organization, and User Interaction. The MPEG-7 DSs can be considered as a library of description tools and, in practice, an application should select an appropriate subset of relevant DSs. This section discusses each of the different functional areas of Multimedia DSs.

2.1.1 Basic Elements

MPEG-7 provides a number of Schema Tools that assist in the formation, packaging, and annotation of MPEG-7 descriptions. An MPEG-7 description begins with a root element that signifies whether the description is complete or partial. A complete description provides a complete, standalone description of AV content for an application. On the other hand, a description unit carries only partial or incremental information that possibly adds to an existing description. In the case of a complete description, an MPEG-7 top-level element follows the root element. The top-level element orients the description around a specific description task, such as the description of a particular type of AV content (for instance an image, video, audio, or multimedia), or a particular function related to content management, (such as creation, usage, summarization, and so forth). The top-level types collect together the appropriate tools for carrying out the specific description task.

In the case of description units, the root element can be followed by an instance of an arbitrary MPEG-7 DS or Descriptor. Unlike a complete description which usually contains a "semantically-complete" MPEG-7 description, a description unit can be used to send a partial description as required by an application such as a description of a place, a shape and texture descriptor and so on. It is also used to define elementary piece of information to be transported or streamed in case the complete description is too large.

A number of basic elements are used as fundamental constructs in defining the MPEG-7 DSs. The basic data types provide a set of extended data types and mathematical structures such as vectors and matrices, which are needed by the DSs for describing AV content. The basic elements include also constructs for linking media files, localizing pieces of content, and describing time, places, persons, individuals, groups, organizations, textual annotation (including free text, structured annotation or annotation with syntactic dependency, etc.), classification schemes and controlled terms.

2.1.2 Content Management

MPEG-7 provides DSs for AV content management. These tools describe the following information: (1) creation and production, (2) media coding, storage and file formats, and (3) content usage.

The Creation Information provides a title (which may itself be textual or another piece of AV content), and information such as creators, creation locations, and dates. It also includes classification information describing how the AV material may be categorized into genre, subject, purpose, language, and so forth. It provides also review and guidance information such as age classification, parental guidance, and subjective review.

The Media Information describes the storage media including format, compression, and coding of the AV content. The Media Information identifies the master media, which is the original source from which different instances of the AV content are produced. The instances of the AV content are referred to as Media Profiles, which are versions of the master obtained by using different encodings, or storage and delivery formats. Each Media Profile is described individually in terms of the encoding parameters, storage media information and location.

The Usage Information describes usage rights, usage record, and financial information. The rights information is not explicitly included in the MPEG-7 description, instead, links are provided to the rights holders and to other information related to rights management and protection.

2.1.3 Content Description: Structural Aspects

The description of the structure of the AV content relies on the notion of segments. The Segment DS describes the result of a spatial, temporal, or spatio-temporal partitioning of the AV content. It can describe a hierarchical decomposition resulting in a segment tree. Moreover, the SegmentRelation DS describes additional relationships among segments and allows the creation of graphs.

The Segment DS forms the base type of the different specialized segment types such as audio segments, video segments, audio-visual segments, moving regions, and still regions. As a result, a segment may have spatial and/or temporal properties. For example, the AudioSegment DS can describe a temporal interval of an audio sequence. The VideoSegment DS describes a set of video frames. The AudioVisualSegment DS describes a combination of audio and visual information such as a video with synchronized audio. The StillRegion DS describes a region of an image or a frame in a video. Finally, the MovingRegion DS describes a moving region of a video sequence.

There exists also a set of specialized segments for specific type of AV content. For example, the Mosaic DS is a specialized type of StillRegion. It describes a mosaic or panoramic view of a video segment [9]. The VideoText and the InkSegment DSs are two subclasses of the MovingRegion DS. The VideoText DS describes a region of video content corresponding to text or captions. This includes superimposed text as well as text appearing in scene. The InkSegment DS describes a segment of an electronic ink document created by a pen-based system or an electronic white-board.

The Segment DS contains elements and attributes that are common to the different segment types. Among the common properties of segments is information related to creation, usage, media location, and text annotation. The Segment DS can be used to describe segments that are not necessarily connected, but composed of several non-connected components. Connectivity refers here to both spatial and temporal domains. A temporal segment (VideoSegment, AudioSegment and AudioVisualSegment) is said to be temporally connected if it is a sequence of continuous video frames or audio samples. A spatial segment (StillRegion) is said spatially connected if it is a group of connected pixels. A spatio-temporal segment (MovingRegion) is said spatially and temporally connected if the temporal segment where it is instantiated is temporally connected and if each one of its temporal instantiations in frames is spatially connected (Note that this is not the classical connectivity in a 3D space).

Fig. 4 illustrates several examples of temporal or spatial segments and their connectivity. Fig. 4.a) and b) illustrate a temporal and a spatial segment composed of a single connected component. Fig. 4.c) and d) illustrate a temporal and a spatial segment composed of three connected components. Note that, in all cases, the Descriptors and DSs attached to the segment are global to the union of the connected components building the segment. At this level, it is not possible to describe individually the connected components of the segment. If connected components have to be described individually, then the segment has to be decomposed into various sub-segments corresponding to its individual connected components.

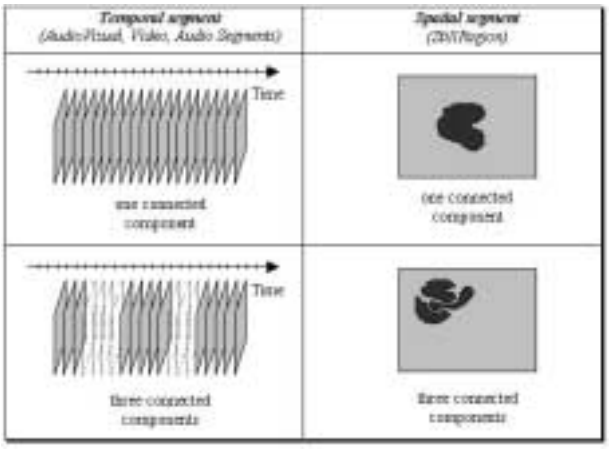


Figure 4: Examples of segments: a) and b) segments composed of one single connected component; c) and d) segments composed of three connected components

The Segment DS may be subdivided into sub-segments, and thus may form a hierarchy (tree). The resulting segment tree is used to describe the media source, the temporal and / or spatial structure of the AV content. For example, a video program may be temporally segmented into various levels of scenes, shots, and micro-segments. A table of contents may thus be generated based on this structure. Similar strategies can be used for spatial and spatio-temporal segments.

A segment may also be decomposed into various media sources such as various audio tracks or viewpoints from several cameras. The hierarchical decomposition is useful to design efficient search strategies (global search to local search). It also allows the description to be scalable: a segment may be described by its direct set of Descriptors and DSs, but it may also be described by the union of the Descriptors and DSs that are related to its sub-segments. Note that a segment may be subdivided into sub-segments of different types, e.g. a video segment may be decomposed in moving regions that are themselves decomposed in still regions.

The decomposition is described by a set of attributes defining the type of sub-division: temporal, spatial, spatio-temporal or media source. Moreover, the spatial and temporal subdivisions may leave gaps and overlaps between the sub-segments. Several examples of decompositions are described for temporal segments in Fig. 5. Fig. 5.a) and b) describe two examples of decompositions without gaps nor overlaps (partition in the mathematical sense). In both cases the union of the children corresponds exactly to the temporal extension of the parent, even if the parent is itself non connected (see the example of Fig. 5.b). Fig. 5.c) shows an example of decomposition with gaps but no overlaps. Final-

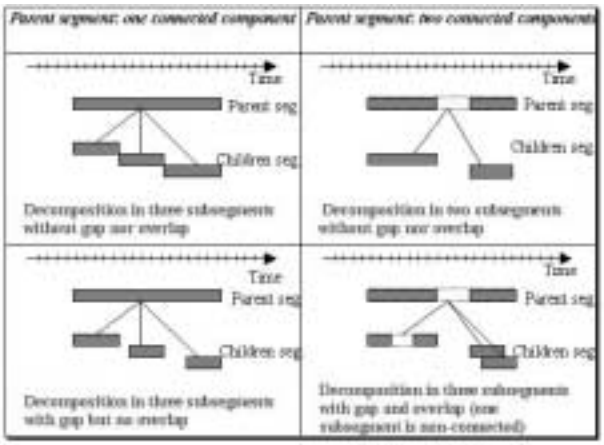


Figure 5: Examples of Segment Decomposition: a) and b) Segment Decompositions without gap nor overlap; c) and d) Segment Decompositions with gap or overlap.

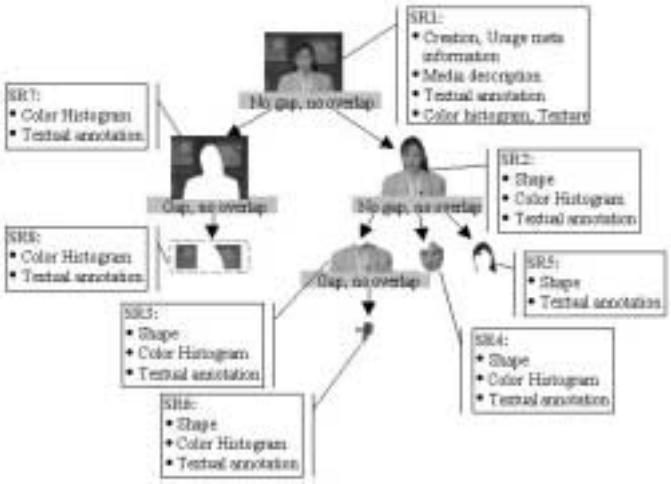


Figure 6: Examples of Image description with Still Regions.

ly, Fig. 5.d) illustrates a more complex case where the parent is composed of two connected components and its decomposition creates three children: the first one is itself composed of two connected components, the two remaining children are composed of a single connected component. The decomposition allows gap and overlap. Note that, in any case, the decomposition implies that the union of the spatio-temporal space defined by the children segments is included in the spatio-temporal space defined by their ancestor segment (children are contained in their ancestors).

As described above, any segment may be described by creation information, usage information, media information and textual annotation. However, specific low-level features depending on the segment type are also allowed. An example of image description is illustrated in Fig. 6. The original image is described as a StillRegion, SR_1, which is described by creation (title, creator), usage information (copyright), media information (file format) as well as a textual annotation (summarizing the image content), a color histogram and a texture descriptor. This initial region can be further decomposed into individual regions. For each decomposition step, we indicate if Gaps and Overlaps are present. The segment tree is composed of 8 StillRegions (note that SR_8 is a single segment made of two connected components). For each region, Fig. 6 shows the type of feature that is instantiated. Note that it is not necessary to repeat in the tree hierarchy the creation, usage information, and media information, since the child segments are assumed to inherit their parent value (unless re-instantiated).

The description of the content structure is not constrained to rely on trees. Although, hierarchical structures such as trees are adequate for efficient access, retrieval and scalable description, they imply constraints that may make them inappropriate for certain applications. In such cases, the SegmentRelation DS has to be used. The graph structure is defined very simply by a set of nodes, each corresponding to a segment, and a set of edges, each corresponding to a relationship between two nodes.

2.1.4 Content Description: Conceptual Aspects

For some applications, the viewpoint described in the previous section is not appropriate because it highlights the structural aspects of the content. For applications where the structure is of no real use, but where the user is mainly interested in the semantic of the content, an alternative approach is provided by the Semantic DS. In this approach, the emphasis is not on Segments but on Events, Objects in narrative worlds and Abstraction. As shown in Fig. 7, the SemanticBase DS describes narrative worlds and semantic entities in a narrative world. In addition, a number of specialized DSs are derived from the generic SemanticBase DS, which describe specific types of semantic entities, such as narrative worlds, objects, agent objects, events, places, time and abstractions.

As in the case of the Segment DS, the conceptual aspects of description can be organized in a tree or in a graph. The graph structure is defined by a set of nodes, representing semantic notions, and a set of edges specifying the relationship between the nodes. Edges are described by the SemanticRelation DSs.

2.1.5 Navigation and Access

MPEG-7 facilitates navigation and access of AV content by describing summaries, views and variations. The Summary DS describes semantically meaningful summaries and abstracts of AV content. The summary descriptions allow the AV content to be navigated in either a hierarchical or sequential fashion. The HierarchicalSummary DS describes the organization of summaries into multiple levels of detail. The main navigation mode is from coarse to fine and vice-versa. Note that the hierarchy may be based on the quantity of information (for example, a few key-frames for a coarse rep-

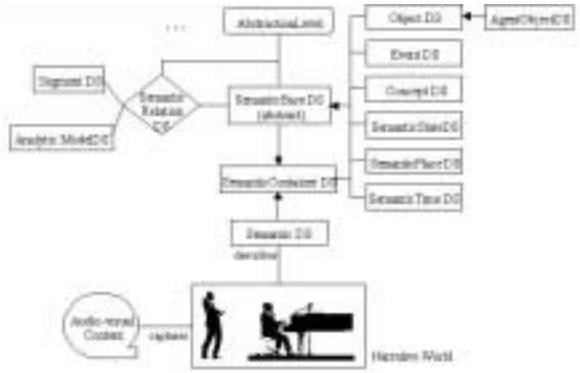


Figure 7: Tools for the description of conceptual aspects.

resentation versus a large number of key-frames for a fine representation) or on specific features (for example, only the most important events are highlighted in the coarse representation whereas a large number of less important events may be shown in the fine representation).

The SequentialSummary DS describes a summary consisting of a sequence of images or video frames, which is possibly synchronized with audio. The SequentialSummary may also contain a sequence of audio clips. The main navigation mode is linear (forward - backward).

The View DS describes structural views of the AV signals in the space or frequency domain in order to enable multiresolution access and progressive retrieval.

Finally, the Variation DS describes relationships between different variations of AV programs. The variations of the AV content include compressed or low-resolution versions, summaries, different languages, and different modalities, such as audio, video, image, text, and so forth. One of the targeted functionalities is to allow a server or proxy to select the most suitable variation of the AV content for delivery according to the capabilities of terminal devices, network conditions, or user preferences.

2.1.6 Content Organization

The Content Organization is built around two main DSs: The Collection DS and the Model DS. The Collection DS includes tools for describing collections of AV material, collections of AV content descriptions, collections of semantic concepts, mixed collections (content, descriptions, and concepts) and collection structures in terms of the relationships among collections.

The Model DS describes parameterized models of AV content, descriptors, or collections. The ProbabilityModel DS describes different statistical functions and probabilistic structures, which can be used to describe samples of AV content and classes of Descriptors using statistical approximation. The AnalyticModel DS describes a collection of examples of AV content or clusters of Descriptors that are used to provide a model for a particular semantic class. For example, a collection of art images labeled with tag indicating that the paintings are examples of the Impressionist period forms an analytic model. The AnalyticModel DS also optionally describes the confidence in which the semantic labels are assigned. The Classifier DS describes different types of classifiers that are used to assign the semantic labels to AV content or collections.

2.1.7 User Interaction

The UserInteraction DS describes preferences of users pertaining to the consumption of the AV content, as well as usage history. The MPEG-7 AV content descriptions can be matched to the preference descriptions in order to select and personalize AV content for more efficient and effective access, presentation and consumption. The UserPreference DS describes preferences for different types of content and modes of browsing, including context dependency in terms of time and place. The UsageHistory DS describes the history of actions carried out by a user of a multimedia system. The usage history descriptions can be exchanged between consumers, their agents, content providers, and devices, and may in turn be used to determine the user's preferences with regard to AV content.

2.2 Visual features

The low-level visual features described in MPEG-7 are color, texture, shape & localization, motion and low-level face characterization. With respect to the Multimedia DSs described in section 2.1, the Descriptors or DSs that handle low-level visual features are to be considered as a characterization of segments. Not all Descriptors and DSs are appropriate for all segments and the set of allowable Descriptors or DSs for each segment type is defined by the standard. This section summarizes the most important description tool dealing with low-level visual features.

2.2.1 Color Feature

MPEG-7 has standardized eight color Descriptors: Color space, Color quantization, Dominant colors, Scalable color histogram, Color structure, Color layout and GoF/GoP color. The first two Descriptors, Color space and quantization, are intended to be used in conjunction with other color Descriptors. Possible color spaces include $\{R, G, B\}$, $\{Y, C_r, C_b\}$, $\{H, S, V\}$, Monochrome and any linear combination of $\{R, G, B\}$. The color quantization supports linear and non-linear quantizers as well as lookup-tables.

The DominantColor Descriptor is suitable for representing local features where a small number of colors are enough to characterize the color information in the region of interest. It can also be used for whole images. The percentage of each color in the region of interest and, optionally, the spatial coherency are described. This Descriptor is mainly used in retrieval by similarity.

The ScalableColorHistogram Descriptor represents a color histogram in the $\{H, S, V\}$ color space. The histogram is encoded with a Haar transform to provide scalability in terms of bin numbers and accuracy. It is particularly attractive for image-to-image matching and color-based retrieval.

The ColorStructure descriptor captures both color content and its structure. Its main functionality is image-to-image matching. The extraction method essentially computes the relative frequency of 8x8 windows that contain a particular color. Therefore, unlike a color histogram, this descriptor can distinguish between two images in which a given color is present with the same probability but where the structures of the corresponding pixels are different.

The ColorLayout descriptor specifies the spatial distribution of colors for high-speed retrieval and browsing. It targets not only image-to-image matching and video-clip-to-video-clip matching, but also layout-based retrieval for color, such as sketch-to-image matching which is not supported by other color descriptors. The Descriptor represents the DCT values of an image or a region that has been previously partitioned into 8x8 blocks and where each block is represented by its dominant color.

The last color Descriptor is the GroupOfFrames/GroupOfPicturesColor descriptor. It extends the ScalableColorHistogram Descriptor defined for still images to video sequences or collection of still images. The extension describes how the individual histograms computed for each image have been combined: by average, median or intersection.

2.2.2 Texture Feature

There are three texture descriptors: Homogeneous Texture, Texture Browsing and Edge Histogram. Homogeneous texture has emerged as an important visual primitive for searching and browsing through large collections of similar looking patterns. The HomogeneousTexture Descriptor provides a quantitative representation. The extraction relies on a frecuencial decomposition with a filter bank based on Gabor functions. The frequency bands are defined by a scale parameter and an orientation parameter. The first and second moments of the energy in the frequency bands are then used as the components of the Descriptor. The number of filters used is 5x6 = 30 where 5 is the number of scales and 6 is the number of orientations used in the Gabor decomposition.

The TextureBrowsing Descriptor provides a qualitative representation of the texture similar to a human characterization, in terms of dominant direction, regularity, and coarseness. It is useful for texture-based browsing applications. The Descriptor represents one or two dominant directions and, for each dominant direction, the regularity (four possible levels) and the coarseness (four possible values) of the texture.

The EdgeHistogram Descriptor represents the histogram of five possible types of edges, namely four directional edges and one non-directional edge. The Descriptor primarily targets image-to-image matching (query by example or by sketch), especially for natural images with non-uniform edge distribution.

2.2.3 Shape and Localization Features

There are five shape or localization descriptors: Region-based Shape, Contour-based Shape, Region Locator, Spatio-temporal Locator, and 3D shape.

a) b) c)

Figure 8: Illustration of Region and Contour similarity

The Region-based and Contour-based Shape Descriptors are intended for shape matching. They do not provide enough information to reconstruct the shape nor to define its position in the image. Two shape Descriptors have been defined because, in terms of applications, there are at least two major interpretations of shape similarity. For example, the shapes represented in Fig. 8.a) and b) are similar because they correspond to a cross. The similarity is based on the contours of the shape and, in particular, on the presence of points of high curvature along the contours. This type of similarity is handled by the Contour-based Shape Descriptor. Shapes illustrated in Fig. 8.b) and c) can also be considered as similar. However, the similarity does not rely on the contours but on the distribution of pixels belonging to the region. This second similarity notion is represented by the Region-based Shape Descriptor.

The Contour-based Shape Descriptor captures characteristics of a shape based on its contour. It relies on the so-called Curvature Scale-Space [8] representation, which captures perceptually meaningful features of the shape. The Descriptor essentially represents the points of high curvature along the contour (position of the point and value of the curvature). This representation has a number of important properties, namely: It captures characteristic features of the shape, enabling efficient similarity-based retrieval. It is robust to non-rigid deformation and partial occlusion.

The Contour-based Shape Descriptor captures the distribution of all pixels within a region. Note that, in contrast with the Contour-based Shape Descriptor, this descriptor can deal with regions made of several connected components or including holes. The Descriptor is based on an *Angular Radial Transform, (ART)* which is a 2D complex transform defined with polar coordinates on the unit disk. The ART basis functions are separable along the angular and radial dimensions. Twelve angular and three radial basis functions are used. The Descriptor represents the set of coefficients resulting from the projection of the binary region into the 36 ART basis functions.

The RegionLocator and the Spatio-temporalLocator combine shape and localization information. Although they may be less efficient in terms of matching for certain applications, they allow the shape to be (partially) reconstructed and positioned in the image. The RegionLocator Descriptor represents the region with a compact and scalable representation of a bounding Box or Polygon. The Spatio-temporalLocator has the same functionality but describes moving regions in a

video sequence. The Descriptor specifies the shape of a region within one frame together with its temporal evolution based on motion.

3DShape information can also be described in MPEG-7. Most of the time, 3D information is represented by polygonal meshes. The 3D shape Descriptor provides an intrinsic shape description of 3D mesh models. It exploits some local attributes of the 3D surface. The Descriptor represents the 3D mesh shape spectrum, which is the histogram of the shape indexes [6] calculated over the entire mesh. The main applications targeted by this Descriptor are search, retrieval and browsing of 3D model databases.

2.2.4 Motion Feature

There are four motion Descriptors: camera motion, object motion trajectory, parametric object motion, and motion activity. The CameraMotion Descriptor characterizes 3-D camera motion parameters. It supports the following basic camera operations: fixed, tracking (horizontal transverse movement, also called traveling in the film industry), booming (vertical transverse movement), dollying (translation along the optical axis), panning (horizontal rotation), tilting (vertical rotation), rolling (rotation around the optical axis) and zooming (change of the focal length), The Descriptor is based on time intervals characterized by their start time, and duration, the type(s) of camera motion during the interval, and the focus-of-expansion (FOE) (or focus-of-contraction FOC). The Descriptor can describe a mixture of different camera motion types. The mixture mode captures globally information about the camera motion parameters, disregarding detailed temporal information.

The MotionTrajectory Descriptor characterizes the temporal evolution of key-points. It is composed of a list of key-points (x,y,z,t) along with a set of optional interpolating functions that describe the trajectory between key-points. The speed is implicitly known by the key-points specification and the acceleration between two key-points can be estimated if a second order interpolating function is used. The key-points are specified by their time instant and their 2-D or 3-D Cartesian coordinates, depending on the intended application. The interpolating functions are defined for each component x(t), y(t), and z(t) independently. The Description is independent of the spatio-temporal resolution of the content (e.g., 24 Hz, 30 Hz, 50 Hz, CIF, SIF, SD, HD, etc.). The granularity of the descriptor is chosen through the number of key-points used for each time interval.

Parametric motion models have been extensively used within various image processing and analysis applications. The ParametricMotion Descriptors defines the motion of regions in video sequences as a 2D parametric model. Specifically, affine models include translations, rotations, scaling and combination of them. Planar perspective models make possible to take into account global deformations associated with perspective projections. Finally, quadratic models makes it possible to describe more complex movements. The parametric model is associated with arbitrary regions over a specified time interval. The motion is captured in a compact manner as a reduced set of parameters.

A human watching a video or animation sequence perceives it as being a “slow” sequence, a “fast paced” sequence, an “action” sequence, etc. The MotionActivity Descriptor captures this intuitive notion of “intensity of action” or “pace of action” in a video segment. Examples of high activity include scenes such as “scoring in a basketball game”, “a high speed car chase” etc. On the other hand, scenes such as “news reader shot” or “an interview scene” are perceived as low action shots. The MotionActivity Descriptor is based of five main features: the intensity of the motion activity (value between 1 and 5), the direction of the activity (optional), the spatial localization, the spatial and the temporal distribution of the activity.

2.2.5 Face Descriptor

The FaceRecognition Descriptor can be used to retrieve face images that match a query face image. The Descriptor is based on the classical eigen faces approach [7]. It represents the projection of a face region onto a set of basis vectors (49 vectors) which span the space of possible face vectors.

3 CHALLENGES FOR VISUAL INFORMATION ANALYSIS

As mentioned in the introduction, the scope of the MPEG-7 standard is to define the syntax and semantics of the DSs and Descriptors. The description generation and consumption are out of the scope of the standard. In practice, this means that feature extraction, indexing process, annotation and authoring tools as well as search & retrieval engines, filtering and browsing devices are non-normative parts of the standard and can lead to future improvements. It has to be mentioned however, that, for low-level features, the distinction between the definition of the semantics of a tool and its extraction may become fuzzy. A typical example is represented by the HomogeneousTexture Descriptor (see section 2.2.2). In order to support interoperability, MPEG-7 has defined the set filters to be used in the decomposition (Gabor filters and their parameters). Beside the implementation, this leaves little room for future studies and improvements. A similar situation can be found for most visual Descriptors described in section 2.2: the definition of their semantics defines partially the extraction process. The main exceptions are the TextureBrowsing and the MotionActivity Descriptors. Indeed, the characterization of the “Texture regularity” or of the “Motion intensity” is qualitatively done. The CameraMotion Descriptor is a special case, because either one has access to the real parameters of the camera or one has to estimate the camera motion from the observed sequence.

The definition of a low-level Descriptor may also lead to the use of a natural matching distance. However, the standardization of matching distances is not considered as being necessary to support interoperability and the standard only provides informative sections in this area. This will certainly be a challenging area in the future.

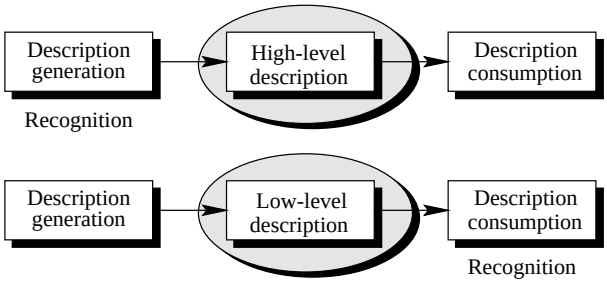


Figure 9: Localization of the recognition process depending on the feature types.

Most of the Descriptors corresponding to low-level features can be extracted automatically from the original content. Most of the time, the main issue is to define the temporal interval or the region of interest that has to be characterized by the Descriptor. This is a classical segmentation problem for which, a large number of tools have been reported in the literature (see [3, 1, 2] and the references herein). An area which has been less worked out is the instantiation of the decomposition involved in the Segment DS. It can be viewed as a hierarchical segmentation problem where elementary entities (region, video segment, and so forth) have to be defined and structured by inclusion relationship within a tree. This process leads, for example, to the extraction of *Tables of Contents* or *Indexes* from the AV content. Although some preliminary results have been reported in the literature, this area still represents a challenge for the future.

One of the most challenging aspects of the MPEG-7 standard in terms of application is to use it efficiently. The selection of the optimum set of DSs and Descriptors for a given application is an open issue. Even if the identification of the basic features that have to be represented is a simple task, the selection of specific descriptors may not be straightforward: for example, DominantColor versus ScalableColorHistogram or MotionTrajectory versus ParametricMotion, etc. Moreover, the real description power of the standard will be obtained when DSs and Descriptors are jointly used and when the entire description is considered as a whole, for example taking into account the various relationships between segments in trees or graphs.

In terms of research, one of the most challenging issues may be the mapping between low-level and high-level descriptions. Let us first discuss the relation between low-level, high-level descriptions and recognition processes. Consider the two situations represented in Fig. 9: on the top, the description is assumed to rely mainly on high-level features. This implies that the automatic or manual indexing process has performed a recognition step during description generation. This approach is very powerful but not very flexible. Indeed, if during the description generation, the high-level feature of interest for the end user has been identified, then the matching and retrieval will be very easy to do. However, if the end user relies on a feature that has not been recognized during the indexing phase, then it is extremely difficult to do anything. The alternative solution is represented in the lower part of Fig. 9. In this case, we assume that the description relies mainly on low-level features. No recognition process is required during the description generation. However, for many applications, the mapping between low-level descriptions and high-level queries will have to be done during the description consumption. That is the search engine or the filtering device will have to analyze the low-level features and, on this basis, perform the recognition process. This is a very challenging task for visual analysis research. Today, the technology related to intelligent search / filtering engines using low-level visual features, possibly together with high-level features, is still very limited. As a final remark, let us mention that this challenging issue has also some implications for the description generation. Indeed, a major open question is to know what are the useful set of low-level Descriptors that have to be used to allow a certain class of recognition tasks to be performed on the description itself.

References

[1] Special issue on segmentation, description and retrieval of video content. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), September 1998.

[2] Special issue on object-based video coding and description. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8), December 1999.

[3] Special issue on image and video processing for digital libraries. *IEEE Transactions on Image Processing*, 9(1), January 2000.

[4] Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, and Eve (Eds.) Maler. XML: Extensible markup language 1.0 (second edition), October 2000. http://www.w3.org/TR/REC-xml.

[5] David C. Fallside (Ed.). XML Schema part 0: Primer, March 2001. http://www.w3.org/TR/xmlschema-0/.

[6] J. Koenderink and A. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8):557–565, 1992.

[7] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, July 1997.

[8] F. Mokhtarian and A.K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.

[9] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:814–830, August 1996.

The Application Model of the MPEG-7 Reference Software - Key Applications and Real World Applications

Stephan Herrmann, Ulrich Niedermeier, and Walter Stechele
Institute for Integrated Circuits, Munich University of Technology,
Arcisstr. 21. 80290 Muenchen, GERMANY
Tel: +49 89 289 25287; fax: +49 89 289 28323
e-mail: stephan.herrmann@ei.tum.de

ABSTRACT

This paper describes the structures of the applications, which are implemented in the MPEG-7 reference software. These applications are basic, and elementary, thus, we call them key applications. After introducing all the key applications implemented in the reference software, we will give a definition for them. Basing on this definition, we can create an abstract model for the key application. This model enables the identification of new possible key applications of the MPEG-7 reference software. We also show how this key application model can be used to describe the relation between the applications of the reference software and real world applications.

1 Introduction

The working item MPEG-7 [1][2] of the Motion Picture Experts Group[1] (MPEG), provides a standard for the multimedia content description interface. Compared to previous MPEG standards (MPEG-1, MPEG-2, MPEG-4) which dealt which multimedia compression technology, MPEG-7 allows to structure, and search in multimedia databases. This is done by defining descriptors for low level features of multimedia content [3][4], and description schemes for high level information [5] of the media content. This process can also be interpreted as some kind of compression. This way, MPEG-7 reduces the information to a more or less semantic level.

Part 6 of the emerging MPEG-7 standard is the MPEG-7 reference software [6], also called the MPEG-7 eXperimentation Model (XM) software. Its purpose is to give an executable specification for the normative components of the standard, which are

- descriptors (Ds) and
- description schemes (DSs),
- the description definition language (DDL) [7], including schema definitions and description instantiations,
- the binary format for MPEG-7 (BiM) [8], which is a binary representation of the DDL, and
- coding schemes (CSs) [3].

In addition to the normative components which are in MPEG-7 in most cases data structures, also procedural, non-normative components show how to extract the descriptions (instances of Ds and DSs) and how to use the extracted descriptions in, e.g., a search & retrieval application. Thus, the XM software supports the written document of the emerging international standard MPEG-7.

Due to the fact, that the XM software implements the normative Ds and DSs, it is also possible to use the XM as a platform to perform core experiments (CEs), which are in the development process of MPEG-7 the essential task to legitimate the standardization of individual techniques. Practically, this was only done in a few cases because

- the XM software integration effort was started after the first set of core experiments was started, and
- some of the core experiments used a complex application to demonstrate the usefulness of the technique.

In section 2 we describe the applications which are implemented in the MPEG-7 reference software. Basing on the individual applications, a more abstract key application model is derived in section 3. This model allows us to describe in section 4 the relation of the XM software applications to real world applications for the MPEG-7 standard.

2 Applications supported by the MPEG-7 reference software

2.1 Extraction vs. Client Applications

The MPEG-7 XM software supports two different types of applications. These are

- extraction applications, and
- client applications.

[1] Working Group 11 form the International Standardization Organization (ISO)

Within the XM software framework, applications are related to one particular descriptor or description scheme. Because there are a lot of descriptors and description schemes standardized, there are also a lot of applications integrated in the software framework. Applications, that are creating the descriptor (D) or description scheme (DS) they are testing, are called extraction applications. On the other hand, applications, which are using the D or DS under test (DUT), are called client applications. Extraction applications are needed if the D or DS is a low level descriptor, which means that the description can be extracted from the multimedia content applying an automatic process. For high level Ds or DSs the extraction cannot be made in an automatic way. However, in most cases the extraction can be done based on preprocessed information. This means, that the extraction process reads this additional information besides the media data to populate the descriptions. Thus, the multimedia content set is extended by additional high level input data.

2.2 Modularity of the XM-software

By default the modules for all Ds and DSs are compiled to build one big executable which can then call the applications for an individual D or DS. However, the resulting executable becomes extremely big, because a lot of individual Ds and DSs are covered by the standard. Compiling the complete framework into one program results in an executable of more than 100 MBytes of size (in case debugging information is enabled). Therefore, the MPEG-7 XM software is designed in a way, that it supports partial compilation to allow to use only one single D or DS. On the other hand, in many cases it is desired to combine a subset of Ds or DSs. Furthermore, combining Ds and DSs is also required in case a DS is built in an hierarchical way from other Ds and DSs. In this scenario, it is not only important to allow partial compilation, but it is essential to design the software to allow as far as possible the reuse of code. As a conclusion, all applications are built from modules. These modules are:

- the media decoder class,
- the multimedia data class,
- the extraction tool class (only for extraction applications),
- the descriptor class,
- the coding scheme class, and
- the search tool class (only for client applications).

To increase the reusability, all this classes are using specified interfaces which are independent from the D or DS they belong to. Thus, it should be possible to reuse ,e.g., the extraction tool of a D or DS in an other D or DS without knowing very deeply what is done in the included extraction tool. This is only possible if it is known how to use the interface of this extraction tool.

The modules listed above are combined or connected to each other to form a processing chain. This is done in the application class. As described in section 2.1, these application classes can be of the extraction- or client application type.

The next part of the section gives a brief description of the listed modules:

2.2.1 Media decoders

The media decoder (MediaIO class) supports a wide range of possible input media formats. These are:

- audio data in WAV files,
- MPEG-1 video streams,
- motion vectors from MPEG-1 video streams (treated as still images),
- still images (JPEG, GIF, PNM, and many more),
- 4D key point lists (t,x,y,z),
- nD key point lists (t, x[0..n-1]), and
- other proprietary input formats for high level information

For this purpose the MediaIO class uses a set of external libraries which do not belong in all cases to the XM software source code tree. These libraries are

- AF library for audio files, and
- ImageMagick for still images.

A special case are video sequences, because the decoded and uncompressed representation is to big to be held in memory. Therefore, the MediaIO class stores the decompressed images in temporary files, which can then be loaded using the routines for still images. The same mechanism is applied to motion vector information, but here the video sequence decoding is stopped after the motion vectors are available.

Because the MediaIO class is an interface to this libraries, the usage of the external libraries is not needed and not allowed in any other class of the XM software. This enables, e.g., that audio experts use the XM software without the video specific ImageMagick library.

2.2.2 Multimedia data

The MultiMedia class holds the loaded media data in memory. Video sequences are, as described in section 2.2.1, not loaded into memory, but only the single frames of the sequence.

For still images the XM uses a reduced structure of the MoMuSys Vop data structure from the MPEG-4 Verification Model (VM). Key points are stored in a two dimensional linked list, one dimension for the time points

(one frame) containing the second dimension, which includes all key points for this frame. The Audio data structure is not concluded at the time being, but will be available in the near future.

2.2.3 Extraction tool

The extraction tool performs the feature extraction for a single element of the multimedia database. The extraction process is a non-normative tool in the MPEG-7 standard. To perform the feature extraction, the extraction tool receives the references to the media data, which is the input for the extraction, and on the other hand the reference to the description, which stores the results from the extraction process.

Because in case of processing video sequences, it is not possible to provide all input data at the same time, the extraction is performed on a per frame basis. This means, that there are three function to be used for performing the extraction:

- InitExtracting which is called before the first frame is processed,
- StartExtracting which is called in a loop over all frames to extract a part of the description, and
- PostExtracting which is called after all frames were processed. This is required if some part of the description can only be generated after all data was available (e.g., the number of frames in the sequence).

The same interface is used in case audio data is processed. Here, the input data is more or less continuous (having only one sample at a time loaded has no meaning). Thus, the input is cut into time frames, which then can be processed one by one.

Besides the interfaces, the extraction classes have procedural code. In case of image or video extraction tools, the XM software uses the AddressLib [9] which is a generic video processing library to perform the low level image processing tasks.

At the time being, the extraction tool is only be used in the *extraction from media* application type. As we will show later, it would also be possible to extract the D or DS under test from other description data. In this case, the extraction process could be performed with only one function call, i.e., without applying a loop iterating the input data for each time point or period.

2.2.4 Descriptor class

The descriptor classes hold the description data. In the XM software the classes for each D or DS represent directly the normative part of the standard. Besides providing the memory for the description also accessory function for the elements of the descriptions are available.

In the XM software there are two different ways of designing the D or DS class. In case of Visual Ds, this class uses a plain C++ class approach. In all other cases this class is implemented using a generic module, which is called the GenericDS in the XM software. This class is an interface from the C++ XM software to the instantiating DDL parser. Concrete, an XML parser providing the DOM-API[2] is used. Therefore, the GenericDS provides the interface from the XM to the DOM-API parser. The memory management for the description data is done by the DOM parser library. Both approaches can be combined using the functions *ImportDDL* and *ExportDLL* of the C++ implemented descriptor classes.

2.2.5 Coding scheme

The coding scheme includes the normative encoder and decoder for a D or DS. In most cases the coding scheme is defined only by the DDL schema definition. Here, the coding is the dumping of the description to a file and the decoding is the parsing and loading of the description file into memory. The description is stored using the GenericDS class which is a wrapper to the DOM-API. Therefore, we can use the DOM-API parser library for encoding and decoding. Again, this functions are wrapped to the XM, using the GenericDSCS (CS = coding scheme) class. Besides the ASCII representation of the XML file, also a binary representation will be standardized by MPEG-7. This is the so called BiM[3] which is one to one equivalent to the ASCII XML representation. At the time being, the BiM is not integrated into the XM software.

Another approach is also be used in the Visual Group of MPEG-7. Here, each D also has an individual binary representation. This allows to specify the number of bits to be used for coding individual elements of the description. An example could be number of bits being used for coding each bin value of a histogram.

2.2.6 Search tool

As the extraction tool, also the search tool represents a non-normative tool of the standard. It takes at the input one description from the data base, and one description for the query, while the query does not need to be compliant to a normative MPEG-7 D or DS. The search tool navigates through the description and processes the required input date in way that it is useful for the specific application.

Search tools are used in all client applications, which are at the time being the *search & retrieval* application and the *media transcoding* application. In case of a search & retrieval application, the search tool compares the two input descriptions and computes a value for the distance between them. In the media transcoding application also media data are processed, i.e., the media information is modified basing on the description and the query. Because media data is processed, the search

[2] Data Object Model - Application Programming Interface
[3] Binary representation for MPEG-7

tool is called in the transcoding application in a media frame by media frame manner as it is done with the extraction tool.

2.3 Extraction from Media

In this section we describe the application types, which are implemented in the XM software.

The *extraction from media* application is of the extraction application type. Usually, all low level Ds or DSs should have an application class of this type. As shown in figure 1 this application extracts the D/DS under test (DUT) from the media input data. First, the media file is loaded by the media decoder into the multimedia class, i.e., into the memory. In the next step, the description can be extracted from the multimedia class using the extraction tool. Then the description is passed through the encoder and the encoded data is written to a file. This process is repeated for all multimedia files in the media database.

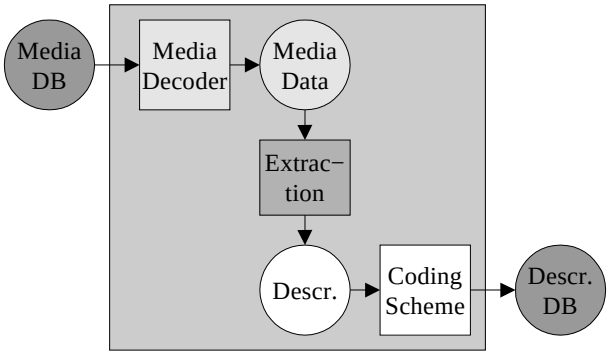


Figure 1: Extraction form media application type. The description is extracted from the media input data.

2.4 Search & Retrieval Application

The search & retrieval application, shown in figure 2 is of the client application type. First all descriptions of the database, which might have been extracted using the *extraction from media* application, are decoded and loaded into the memory. Also the query description can be extracted from media using the extraction tool. On the other hand the query can also be loaded directly from a file. After having all input data, the query is processed on all elements of the database, and the resulting distance values are used to sort the data base with decreasing similarity to the query. Finally, the sorted list is written as a new media database to a file.

Search & retrieval applications are extremely well apt for client applications of low level features. For low level features an evaluation criterion can be formulated because here an important question is how good the description represents the low level feature. The best way to evaluate this criterion is to define a ground truth

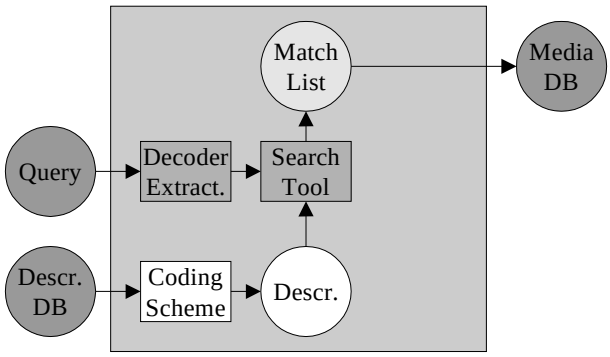


Figure 2: Search & retrieval application type. A sorted media database is created from the descriptions and a query.

dataset for the queries and to check the retrieval rate with the search & retrieval application.

2.5 Media Transcoding Application

The media transcoding application is also of the the client application type. As shown in figure 3, the media files and their description are loaded. Basing on the descriptions, the media data are modified (transcoded), and the new media database is written to a file. Furthermore, a query can be specified, which is processed on the description prior to the transcoding.

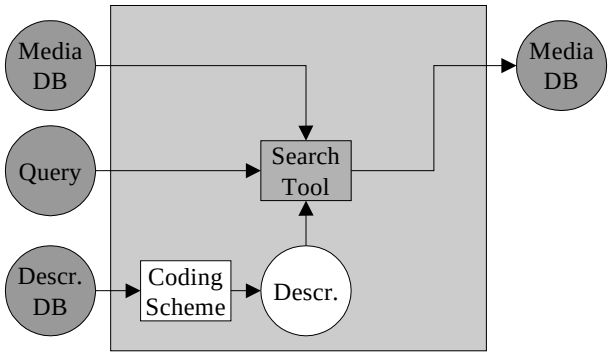


Figure 3: Media transcoding application type. A transcoded media database is created from the original media database, the corresponding descriptions, and an optional query.

3 The MPEG-7 key application model

3.1 Definition of key applications

In the previous section the applications, which are implemented in the XM software, were described. This applications are also called key application, because they are basic or elementary application types. They represent different application scenarios by implementing the key features of this application scenarios. In general,

key applications are not necessarily real world applications because they only implement the representative and common task of the application scenarios. Details characterizing a specific real world application are not implemented.

Another important limitation of the XM software is the fact, that the XM software is a command line tool only, i.e., that the application, its inputs and outputs can only be specified when the XM is started. As a conclusion, the key applications do not support user interaction during run time.

However, the key applications can be defined by the interfaces they are using at the input and output. Table 1 summarizes the interfaces used in the applications that are described in section 2.

Application	Inputs	Outputs
Extraction from Media	media db	description db
Search & Retrieval	description db, query	media db (list of best matches)
Media Transcoding	media db, description db, query	media db (transcoded media)

Table 1: Key applications and their inputs and outputs (media db = media database, description db = description database)

Summarizing, key applications are elementary applications, which can be specified by their interfaces, and having no proprietary application characteristic, especially no specific user interaction.

3.2 The interface model

After identifying the nature of key applications the second step is the design of an abstract key application model. Basing on the definition of key applications which was done using the interfaces, all possible inputs and outputs used by key applications can be collected. The resulting subset of the inputs and outputs is shown in figure 4. Possible inputs are media databases, description databases, and queries. Possible outputs are media databases, and description databases. In the abstract model the semantics of the media database output is not distinguished, i.e, the list of best matching media files and the transcoded media database are not treated as individual output types, but they are in principle of the same kind.

Besides the already used outputs, it is assumed that there will be also a corresponding output type for the query input. In figure 4 this output has the name *other outputs*. Possible applications for this could be a refined query, e.g., for a browsing application. However, the usage of this output is still not clear and needs further investigations.

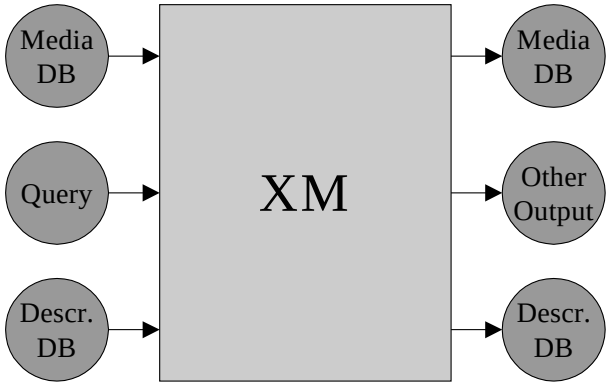


Figure 4: Interface model for XM key applications. This model shows the superset of possible inputs and outputs of an XM key application.

In the following we use the interface model of the key applications for two purposes, which are the identification of new relevant key applications and the description of relations of key applications to real world applications.

3.3 Other possible key applications

The list of existing key applications at the time being is still incomplete. Using the interface model, theoretically eight combinations for input data as well as for output data are possible. One of the eight combinations for the input data uses no input at all. Having no input data would only allow to create results in a random way. Furthermore, using a query without description data has no meaning. On the other hand, also the possible range of output combinations can be limited. Thus, it is possible to decompose an application producing two outputs in two applications producing each one of of the outputs. As a conclusion table 2 shows the relevant combinations of inputs and outputs.

The description filter application might appear in two different ways, as a description filter extraction (creating the D or DS under test (DUT)), or as a description filter client application (reading the DUT at the input).

4 Key applications vs. real world applications

As stated in section 3.1, the key applications in the XM software are elementary application types. In general, combining the key applications will form complex applications. Because the key applications can have arbitrary combinations of inputs, the key application model is generic for this application area. Therefore, it is also possible that real world applications can be decomposed into processing networks consisting of the elementary key application blocks, and user interfaces providing user interaction and presentation of results.

Inputs	Output	application name
	M	Random Search
	D	Random Extraction
	O	not relevant for MPEG-7
M	M	MPEG-7 unrelated transcoding
M	D	Extraction from Media
M	O	not relevant for MPEG-7
D/DQ	M	Search & Retrieval
D/DQ	D	Description Filter
D/DQ	O	unknown
MD/MDQ	M	Media Transcoding
MD/MDQ	D	unknown
MD/MDQ	O	unknown

Table 2: Possible input output combinations for key applications of the XM. (M = media db; D = description db; Q = query; O = other output

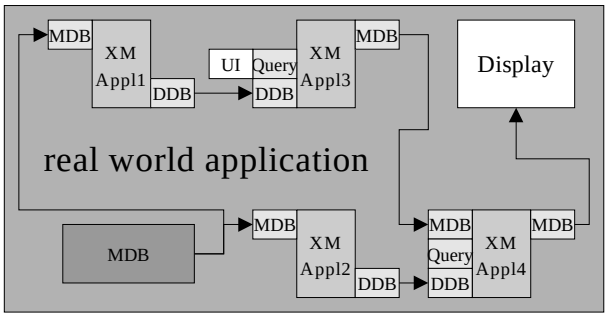


Figure 5: Example real world application extracting two different descriptions (XM-Appl1, XM-Appl2). Basing on the first description the relevant content set is selected (XM-Appl3) which is then transcoded using the second description (XM-Appl4). (MDB = media database, DDB = description database)

Figure 5 shows an example real world application. Here, from a media database two features are extracted. Then, basing on the first feature, relevant media files are selected from the media database. The relevant media files are transcoded basing on the the second extracted feature.

On the one hand, this is helpful for designing applications and products. On the other hand, also core experiments used in the standardization process are done basing on applications, which are in some cases more complex than key applications. In this cases, the decomposition or relation to the key applications can help do define evaluation criteria for the core experiments. The evaluation criteria are clear for the search & retrieval application, which is the retrieval rate, and for the extraction application, which is the computational complexity. In general also the bit stream complexity is an important evaluation criterion for all key applications. At the time being, the evaluation criteria are not clear for all possible key applications. However, answering this question for key applications seams more feasible than for proprietary applications.

5 Conclusions

This paper showed the applications currently implemented in the MPEG-7 reference software. These applications are called key applications. Starting from the individual key applications, an abstract key application model was derived. This model is based on the interfaces which are used by the key applications. The resulting key application model is relevant for the identification of new key applications to complete the set of applications of the reference software, to understand for which key applications evaluation criteria should be identified, and to define the relation between the MPEG-7 reference software and real world applications. The last point is essential for designing and understanding applications and products for the emerging MPEG-7 standard.

References

[1] MPEG Requirements Group, "MPEG-7 Requirements," Singapore: 2001.

[2] MPEG Requirements Group, "MPEG-7 Overview," Singapore: 2001.

[3] ISO/IEC, "Part 3 - MPEG-7 FCD's (Visual)," Singapore: 2001.

[4] ISO/IEC, "Text of ISO/IEC FCD 15938-4 Information Technology - Multimedia Content Description Interface - Part 4 Audio," Singapore: 2001.

[5] ISO/IEC, "Text of ISO/IEC 15938-5 FCD Information Technology - Multimedia Content Description Interface - Part 5 Multimedia Description Schemes," Singapore: 2001.

[6] ISO/IEC, "Text of ISO/IEC FCD 15938-6 Information Technology - Multimedia Content Description Interface- Part 6: Reference Software (Version 1.0)," Singapore: 2001.

[7] ISO/IEC, "Text of ISO/IEC 15938-2/FCD (DDL)," Singapore: 2001.

[8] ISO/IEC, "Text of ISO/IEC 15938-1/FCD (Systems)," Singapore: 2001.

[9] S. Herrmann and H. Mooshofer and W. Stechele, "A Toolbox Approach for Image Segmentation and Complexity Analysis," Proceedings of the Workshop WIAMIS'97, Lovain-la-Neuve, June 1997.

PicSOM CONTENT-BASED IMAGE RETRIEVAL SYSTEM – COMPARISON OF TECHNIQUES

Markus Koskela, Jorma Laaksonen, and Erkki Oja
Laboratory of Computer and Information Science, Helsinki University of Technology,
P.O.BOX 5400, 02015 HUT, FINLAND
e-mail: {markus.koskela,jorma.laaksonen,erkki.oja}@hut.fi

ABSTRACT

The operation of a content-based image retrieval (CBIR) system can be seen as a series of consecutive processing stages. As there exists multiple choices for each of these stages, a multitude of CBIR systems can be implemented by combining a set of common building blocks. In CBIR, images are indexed on the basis of low-level features, such as color, texture, and shape, that can automatically be derived from the visual content of the images. As image similarity is based on such low-level features, the similarity of images is dependent on the used feature extraction scheme. Furthermore, any such single feature is unlikely to provide adequate classification ability for images belonging to miscellaneous semantic categories. An effective CBIR system should therefore support multiple features in parallel and combine their responses in a meaningful manner. In this paper, we present comparison results for a three-stage CBIR system. The results of the performed experiments show that CBIR systems can be implemented using consecutive stages where, at each stage, a number of parallel techniques can be provided.

1 INTRODUCTION

Content-based retrieval from unannotated image databases is a wide and versatile field of research interests. Despite considerable research efforts during the last decade, successful CBIR applications have not yet emerged. The fundamental problem in CBIR is the gap between the high-level semantic concepts used by humans to perceive images and the low-level visual features used by computers to index the images in a database. In addition to developing descriptive visual features, another key issue is the image indexing method, ie., how to select the images the system returns as the result of an image query.

The operation of a CBIR system can be seen as a series of more or less independent processing stages. For each of these stages, complementary algorithms can be developed. As a result, full CBIR systems can be implemented by combining a set of common building blocks. We have implemented such a framework of a CBIR system containing three consecutive stages. These include (1) the initial per-feature selection of considered images, (2) the combination of the lists of images selected in the first stage, and (3) the final selection of images based on all the available features simultaneously. In this paper, four different techniques for the first selection stage, two choices for the combination stage, and three for the final selection stage are evaluated.

As image retrieval systems are usually not capable of giving the wanted images in its first response to the user, the image query becomes an iterative and interactive process towards the desired image or images. This automatic refinement of a query is known as *relevance feedback* in information retrieval literature [8]. An image retrieval system implementing relevance feedback tries to learn the optimal correspondence between the high-level concepts people use and the low-level features obtainable from the images.

2 A GENERAL SYSTEM STRUCTURE

When a CBIR system is implemented with prototype-based statistical methods, each image in the database is transformed with a set of different feature extraction methods to a set of lower-dimensional prototypes in respective feature spaces. When the system tries to find images which are similar to the positive-marked seen images, it searches for those images whose distance to the positive images in some sense is minimal in any or all of the feature spaces. The distances between prototypes in the feature spaces can be defined in a multitude of ways, the Euclidean distance being the one used most.

The CBIR process can to some extent be formalized by denoting the set of images in the database as $\mathcal{D}$ and its non-intersecting subsets of positive and negative seen images as $\mathcal{D}^+$ and $\mathcal{D}^-$, respectively. The unseen images can then be marked as $\mathcal{D}'$, which leads to

$$\mathcal{D}' = \mathcal{D} \setminus (\mathcal{D}^+ \cup \mathcal{D}^-) \quad , \tag{1}$$

$$N' = N - (N^+ + N^-) \quad , \tag{2}$$

where the Ns denote the cardinalities of the respective sets. Let us denote the images as $\mathcal{I}_n$, $n = 1, 2, \ldots, N$. If we have M different feature representations for each

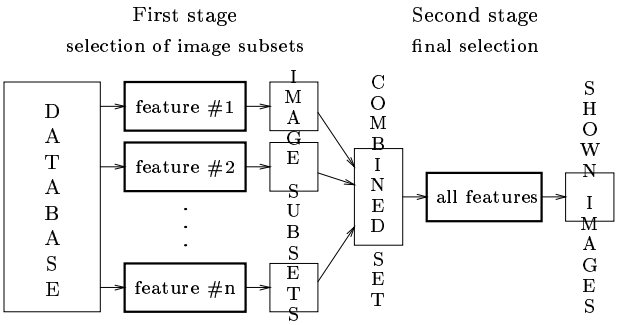


Figure 1: The stages of image selection in CBIR.

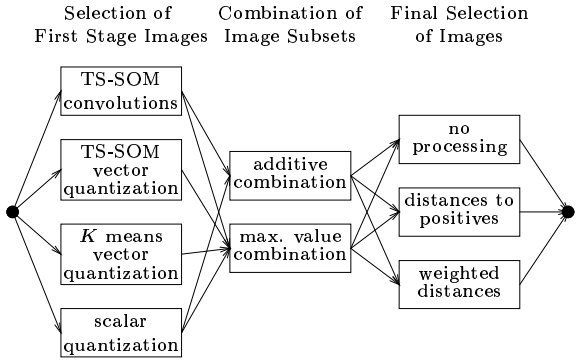


Figure 2: The processing stages in the experiments.

image, they can be written as $\mathbf{f}^m(\mathcal{I}_n) = \mathbf{f}_n^m$, $m = 1, 2, \ldots, M$. The L images the system will display to the user next can be denoted with $\mathcal{D}^* = \{\mathcal{I}_1^*, \mathcal{I}_2^*, \ldots, \mathcal{I}_L^*\} \subset \mathcal{D}'$. Finding the images most similar to the positive seen images can then be formally written, for example, in a straightforward manner:

$$\min_{\mathcal{D}^*}! \; d = \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N^+} w_m \, d_m(\mathbf{f}^m(\mathcal{I}_l^*), \mathbf{f}^m(\mathcal{I}_n^+)) \quad , \quad (3)$$

where w_ms are the weights for individual features and $d_m(\cdot, \cdot)$ is the distance function suitable for being used with feature type $\mathbf{f}^m$. Though (3) is quite general in nature, it is still only one possibility among others.

An image database may contain millions of images. Often it is not possible to calculate accurately all distances between all the positive seen images and all the unseen images in the database and therefore some computational shortcuts need to be taken into account. One approach is to divide and conquer the image selection process by making it in two stages. Figure 1 illustrates this idea. Each feature representation can be used separately for finding a set of image candidates. The number of images in each subset may and should exceed the count of images to be finally shown to the user. These subsets with associated qualification values for each image included should then be combined in a larger set of images which will be processed in a more exhaustive manner. Depending on the sizes of the subsets and the combination algorithm, either all images in them or, for example, only those which are included in more than one of them, can be taken in the combined set.

We have extended our PicSOM [6, 5] CBIR system to accommodate alternative methods in any of the processing stages. For the first selection stage, we have implemented three new techniques that compete with the original method based on convolving the user's response on the Tree Structured Self-Organizing Maps (TS-SOMs). For the combination of the image subsets, we have added a new alternative treatment, namely the use of the maximum of the per-feature qualification values for each image instead of their sum. The original algorithm used in PicSOM does not contain any processing after the combination of qualification values from different features. Now we have implemented two choices for that purpose. Figure 2 displays all the alternative functions for each of the stages.

3 SELECTION OF FIRST STAGE IMAGES

According to the structure illustrated in Figure 1, the first stage of selection is performed for each feature in parallel. In this study, three alternative approaches to this per-feature selection were applied. These approaches are based on the TS-SOM, vector quantization, and scalar quantization, respectively, and are described in this section.

3.1 Tree Structured SOMs

The first image indexing method used in the comparison is based on the Self-Organizing Map (SOM) [3]. The SOM defines an elastic net of points that are fitted to the input space. It can thus be used to visualize multidimensional data, usually on a two-dimensional grid. The SOM consists of a regular grid of neurons where a model vector is associated with each map unit. The map attempts to represent all the available observations with optimal accuracy using a restricted set of models. In order to alleviate the computational complexity of training large SOMs, we use a special form of the SOM, namely the Tree Structured Self-Organizing Map (TS-SOM) [4]. The TS-SOM is used to represent the database in several hierarchical two-dimensional lattices of neurons. Each feature is used separately to train a corresponding TS-SOM structure. As the SOM algorithm organizes similar feature vectors in nearby neurons, the resulting map contains a representation of the database with similar images, according to the given feature, located near each other. The tree structure of the TS-SOM, on the other hand, provides several map levels forming a set of SOMs with different resolutions.

With the TS-SOM algorithm, the system marks the images selected by the user with a positive value and the other images with a negative value in its internal data structure. These values are then summed up in their

best-matching SOM units in each of the TS-SOM maps. Each SOM level is then treated as a two-dimensional matrix formed of values describing the user's responses to the contents of the map unit. Finally, the map matrices are low-pass filtered with Gaussian convolution masks in order to spread the user's responses to the neighboring units which, by presumption, contain images that are to some extent similar to the present ones. Starting from the SOM unit having the largest response after the convolution, the algorithm retrieves the image whose feature vector is nearest to the weight vector in that unit. If that image has not been shown to the user, it is marked to be shown on the next round. This process is continued with the second largest value and so on until a preset number of new images have been selected.

3.2 Vector Quantization

The use of SOMs in indexing images according to their mutual similarity can be regarded as a special case of *vector quantization* (VQ). In the general case of VQ, the topological ordering provided by the map lattice is lost and the similarity of two images is characterized only by whether they are mapped to same quantization bin. The model vectors obtained by training the TS-SOMs can also be used as VQ codebooks. As the vector quantization produced by the TS-SOM is by no means optimal for representing the original data distribution, alternative quantization techniques can also be used. These include the well-known K-means or Linde-Buzo-Gray vector quantization [7]. With either quantization, the feature vectors are divided in subsets in which the vectors resemble each other. Those unseen images which have fallen into the same quantization bins as the positive-marked shown images are then good candidates for the next images to be displayed to the user.

The quantization bins can be divided into the following four categories, in decreasing order of importance. First, bins containing both positive seen images and yet unseen images. These bins form a natural subset to concentrate on when searching for new images and they can be scored by the fraction of positive images in all seen images in them. The second category consists of bins containing only unseen images. The third category is formed of bins whose all seen images have been negative. These bins are sorted by the ratio of unseen images to all images in them. The last category consists of bins having no unseen images. These are of no interest and can be discarded. Consequently, the quantization bins are scored and sorted according to the qualification value

$$S_i = \begin{cases} 1 + \frac{N_i^+}{N_i^+ + N_i^-} & , \text{if } N_i^+ > 0 \wedge N_i' > 0 \\ 1 & , \text{if } N_i^+ = 0 \wedge N_i^- = 0 \wedge N_i' > 0 \\ \frac{N_i'}{N_i} & , \text{if } N_i^+ = 0 \wedge N_i^- > 0 \wedge N_i' > 0 \\ 0 & , \text{otherwise} \end{cases} \quad (4)$$

where N_i^+, N_i^-, and N_i' are the numbers of positive and negative seen images and unseen images, respectively, mapped to vector quantization bin i. N_i is the total number of images in the bin.

From each vector quantization, an image subset of K images is selected in the order of descending S_i. The images from bins containing positive images, ie. the first category in (4), are picked until the limit of K images is reached. If there are not enough images in that bin the picking is continued in the bin with the next largest S_i and so on. If the count K could not be filled from bins with S_i larger than one, the remaining images are picked from the following categories using breadth first search, ie. picking only one image from each bin. This mode of operation is, however, only necessary in the end of an exhaustive query. Each image that is included in the resulting image subset is assigned its quantization bin's qualification value S_i.

3.3 Scalar Quantization

In the case of scalar quantization the resemblance between images is in the first place found with respect to one component of a feature vector. Scalar quantization can be obtained by ordering the values in certain feature vector component. Ordered values can then be divided in a preset number of quantization bins each containing a few images. All images in the bin are then indexed with an inverse file. When the ordering and creation of the inverse is repeated for all components of the particular feature vector, one obtains different scalar quantization for each single feature component.

The similarity of two images can be expressed by the count of feature components for which they are quantized to same bin. With $\mathcal{D}^+$, the set of positive seen images, one can then rank all images which are mapped to the same scalar quantization bin as one or more images of $\mathcal{D}^+$, by counting the overall sum of bins shared by the image in question and the images of $\mathcal{D}^+$.

In order to prevent longer feature vectors from dominating the creation of the combined image set in the following processing stage, we have divided the number of positive quantization bins shared by the image in question with the dimensionality of the feature vector to obtain the qualification value for the image.

4 COMBINATION OF IMAGE SUBSETS

The next step in the system of Figure 1 is combining the per-feature subsets of images provided by the first stage of processing. Mainly, this consists of simply taking an union of the image sets, but there are two issues requiring some consideration, namely dealing with duplicate images and limiting the size of the combined set.

4.1 Additive Value Combination

If the image values provided by the first stage are additive, ie. they provide, in addition to ordinal scale, also information in the differences in size between values, a

natural way to combine the image sets is to use value addition for duplicate images. This rewards images considered promising by multiple features, which, by assumption, is a desirable property. The convolved responses of TS-SOMs and the qualification values produced by scalar quantization fulfill this requirement.

4.2 Maximal Value Combination

Another combination approach, suitable for all image sets, is to use maximal values for duplicate images. This removes the effect of possible strong positive responses from multiple features. Maximal value combination can therefore be considered as a secondary option. The scoring function of vector quantization (4) can be regarded to provide only ordinal values and therefore maximal combination needs to be used with that algorithm.

5 FINAL SELECTION OF IMAGES

The last stage of processing, see Figure 1, is intended for a set of potentially relevant images. Here, the selection algorithm may be totally different from the one used in the first stage. In order to enable more demanding processing techniques, the set of remaining images should at this stage be substantially smaller than the whole database.

5.1 Selection by First Stage Information Only

The simplest processing at this stage is to do nothing. If the first selection stage and the set combination step already provide a justifiable set of images, it can be shown to the user as the system's response.

5.2 Selection by Distances to Positive Images

One possible method for selecting the final set of images is to rank the remaining images based on their cumulative distance to all already found positive-marked, ie., relevant images in the original feature space. With this method, the final selection is thus performed according to (3) with $w_m = 1$ for all features and

$$d_m(\mathbf{f}^m(\mathcal{I}_l^*), \mathbf{f}^m(\mathcal{I}_n^+)) = (\mathbf{f}^m(\mathcal{I}_l^*) - \mathbf{f}^m(\mathcal{I}_n^+))^T(\mathbf{f}^m(\mathcal{I}_l^*) - \mathbf{f}^m(\mathcal{I}_n^+)) \ , \quad (5)$$

ie. the squared Euclidean distance.

As calculating distance in a possibly very high-dimensional space is a computationally heavy operation, it may not be feasible to perform it for all images in a large database. Therefore, the first stage can be seen to act as a preprocessor which prunes the database as much as it is necessary before the actual image similarity assessment is carried out by using (5).

5.3 Selection by Weighted Distances

In the previous method, the weights w_m in (3) were all set to one. These feature weights, however, provide an opportunity for adaptation, as features which seem to work well in a given query could be given greater influence in determining the shown images. One approach to implementing this adaptation is to weight features based on the pairwise distances of the found positive images in the feature space. If the average distance of two positive images is relatively small, the feature can be considered as well-suited for the current image query. The absolute distances in different feature spaces vary and therefore the distances have to be normalized with the average pairwise distance of all images in the database. For a feature m, the weight w_m is thus given by

$$w_m = \frac{N(N-1)}{N^+(N^+-1)} \frac{\sum_{i=1}^{N^+}\sum_{j=i}^{N^+} d_m(\mathbf{f}^m(\mathcal{I}_i), \mathbf{f}^m(\mathcal{I}_j))}{\sum_{k=1}^{N}\sum_{l=k}^{N} d_m(\mathbf{f}^m(\mathcal{I}_k), \mathbf{f}^m(\mathcal{I}_l))} \ . \quad (6)$$

6 EXPERIMENT SETTINGS

In order to evaluate the applicability of the presented methods and their different combinations for CBIR, a series of experiments was performed. In this section, the experiment settings, including the used system framework, the image database, visual features, the generation of ground-truth information, and the performance measure, are described.

6.1 PicSOM

The PicSOM image retrieval system is a framework for research on algorithms and methods for content-based image retrieval. A more detailed description of the system and results of earlier experiments performed using the system can be found in [6, 5]. The PicSOM home page including a working demonstration of the system for public access is located at http://www.cis.hut.fi/picsom.

In PicSOM, the queries are performed through a WWW-based user interface and the queries are iteratively refined as the system exposes more images to the user. PicSOM supports multiple parallel features and the responses from the used features are combined automatically by using the map surface convolutions and additive combination of the qualification values, as described above. The goal is to autonomously adapt to the user's preferences regarding the similarity of images in the database.

6.2 Image Database and Ground-Truth Classes

We evaluated the CBIR approaches with a set of experiments using an image collection from the Corel Gallery 1 000 000 product. The collection contains 59 995 photographs and artificial images with a very wide variety of subjects. All the images are either of size 256×384 or 384×256 pixels. The majority of the images are in color, but there are also a small number of grayscale images.

For the experiments, three separate image classes were picked manually from the database. The selected classes were *cars*, *faces* and *planes*, of which

the database consists of 864, 1115 and 292 images, respectively. The corresponding *a priori* probabilities are 1.4%, 1.9%, and 0.5%. In the retrieval experiments these classes were thus not competing against each other but mainly against the "background" of 57 724, ie., 96.2% of other images.

6.3 Features

The features used in the experiments included two different color and shape features and a simple texture feature. All except one shape-based feature were calculated in five separate zones of the image. The zones were formed by first determining in the center of the image a circular area whose size is one fifth of the area of the whole image. Then the remaining area was divided into four zones with two diagonal lines. For a detailed description of the used features, see [1, 6].

In our previous experiments with the PicSOM system, it was found out that using a larger set of features generally yields better results and that the used approach provides a robust method for using a set of different features in parallel so that the result exceeds the performances of all the single features [6]. Therefore, all five features were used in parallel in all experiments.

6.4 Parameters of the Methods

The TS-SOMs for all the five features were sized 4×4, 16×16, 64×64, and 256×256, from top to bottom. On the bottommost TS-SOM levels there were thus approximately the same number of SOM units (65 536) and database images (59 995) During the SOM training, each vector was used 100 times in the adaptation.

In vector quantization, one possibility is to use the same TS-SOMs but only for vector quantization purposes. Of the four TS-SOM levels we chose to use the second from bottom, ie. the one sized 64×64. On the average there were thus approximately 14 images mapped in each quantization bin. With the K-means vector quantization, the same number of quantization bins was used, which again results to the average of 14 images per bin. In the scalar quantization case, were used 15 000 bins which gives rise to approximately 4 images in each bin.

All the image subsets of Figure 1 contained 100 images before duplicate removal. The resulting combined image list was not shortened but all the images were involved in the final selection. After it, 20 best-scoring images were used as the system's response.

6.5 Performance Measure

In this section, the used performance measure, denoted as the τ measure, is presented. With the τ measure, it is assumed that the user is facing a target search task from $\mathcal{D}$ for an image I belonging to class $\mathcal{C} \subset \mathcal{D}$. Before the correct image is found, the user guides the search by marking all shown images which belong to $\mathcal{C}$ as relevant. This process is then repeated for each image in $\mathcal{C}$. Now, the τ measure equals the average number of images the system retrieves before the correct one is found. The τ measure resembles the "target testing" method presented in [2], but instead of relying on human test users, the τ measure is fully automatic.

The τ measure can be obtained by implementing an "ideal screener", a computer program which simulates the human user by examining the output of the retrieval system and marking the images returned by the system either as relevant (positive) or non-relevant (negative) according to whether the images belong to $\mathcal{C}$. This process is continued until all images in $\mathcal{C}$ have been found. For each of the images in the class $\mathcal{C}$, we then record the total number of images presented by the system until that particular image is shown. From this data, we form a histogram and calculate the average number of shown images needed before a hit occurs. In the optimal case, the system first presents all images in $\mathcal{C}$. The optimal value for the average number of images presented before a particular image in $\mathcal{C}$ is thus $\frac{N_\mathcal{C}}{2}$, where $N_\mathcal{C}$ is the number of images in $\mathcal{C}$.

The τ measure for class $\mathcal{C}$ is then obtained by dividing the average number of shown images by the size of the database, N. The τ measure yields a value

$$\tau \in [\frac{\rho_\mathcal{C}}{2}, 1 - \frac{\rho_\mathcal{C}}{2}] \tag{7}$$

where $\rho_\mathcal{C} = \frac{N_\mathcal{C}}{N}$ is the *a priori* probability of the class $\mathcal{C}$. For values $\tau < 0.5$, the performance of the system is thus better than random picking of images and, in general, the smaller the τ value the better the performance.

7 RESULTS

The results of the experiments are listed in Table 1. The rows of the table contain the first stage alternatives, Tree Structured SOMs, vector quantization using TS-SOM and K-means-based quantization, and scalar quantization. Similarly, the columns contain the methods for final stage processing. The first two columns differ only by the method for combining first stage image sets, addition (first column) and maximal value (second column). Results for using addition-based combining with vector quantization were not computed. In the last two columns, the used image set combination method is the standard one for each first stage method, ie. addition for TS-SOMs and scalar quantization, maximum value for vector quantization. Each entry in the table lists first the average value obtained from using the three image classes. The results for individual classes are shown in parentheses, in the following order: cars, faces, and planes.

First, considering the two first columns of Table 1, corresponding to the first stage, it can be seen that the TS-SOM algorithm yields best values for the τ measure. However, the overall best τ values are obtained using vector quantization with K-means clustering and second stage processing. The result when using TS-SOMs

Table 1: Results of the retrieval experiments using the τ measure. Each entry in the table lists first the average value of using the three image classes and then results for the used classes (cars, faces, planes) in parentheses.

	no 2nd stage, add comb.	no 2nd stage, max comb.	distance to positives	weighted distance
SOM	0.174 (0.177 0.209 0.137)	0.175 (0.177 0.210 0.137)	0.190 (0.193 0.229 0.147)	0.179 (0.195 0.209 0.130)
VQ (SOM)		0.217 (0.212 0.235 0.203)	0.184 (0.187 0.181 0.185)	0.178 (0.191 0.164 0.178)
VQ (K-m)		0.185 (0.173 0.214 0.169)	0.155 (0.148 0.167 0.149)	0.154 (0.154 0.162 0.146)
SQ	0.251 (0.363 0.242 0.147)	0.302 (0.392 0.342 0.172)	0.300 (0.388 0.351 0.162)	0.267 (0.356 0.290 0.156)

for quantization is worse as can be anticipated since the topological information of SOMs is discarded with vector quantization. Furthermore, the vector quantization algorithm can be seen to require the final stage of processing. Of the two final stage techniques, the weighted distance seems to yield better results on the average.

The second-best results are obtained with the TS-SOM-based technique. The best results are located at the first column, corresponding to the standard version in which the final stage is not used. Also, it can be seen that adding this extra processing does not improve the results. More strikingly, using maximum value combination yields very similar results. Also, of the two algoritms applying SOMs, the TS-SOM-based technique performs better.

Of the studied first stage techniques, the performance of scalar quantization is clearly the worst. Another observation here is that the additive combination is notably better than using the maximum value. This may suggest that vector quantization might also benefit from using value addition for duplicates. This requires, however, a different scoring function than the one used in these experiments.

8 CONCLUSIONS AND FUTURE PLANS

In this paper, a general multi-part structure of content-based image retrieval systems was presented. A variety of different CBIR techniques can be represented in terms of this system structure. Also, CBIR systems can be considered as sets of basic building blocks. The performance of different systems can then be obtained by analyzing these blocks.

In the set of experiments, the K-means based vector quantization together with the weighted distance to the positive examples was found to give best results according to the used τ measure. The TS-SOM-based technique yielded second-best results. It should be noted, however, that the topological ordering of the images on the TS-SOM map lattice is an additional, unique benefit which cannot be obtained with traditional vector quantization techniques.

The experiments reported here were performed using a set of non-standard features due to the lack of widely-accepted standards for description of image content. Recently, however, the MPEG-7 effort has matured and we are currently testing the suitability of the MPEG-7-defined image descriptors for our system. By using a standard set of feature descriptors, comparative studies between CBIR systems will become easier to perform.

ACKNOWLEDGMENTS

This work was supported by the Finnish Centre of Excellence Programme (2000-2005) of the Academy of Finland, project New information processing principles, 44886.

REFERENCES

[1] S. Brandt, J. Laaksonen, and E. Oja. Statistical shape features in content-based image retrieval. In *Proceedings of 15th International Conference on Pattern Recognition*, volume 2, pages 1066–1069, Barcelona, Spain, September 2000.

[2] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Target testing and the PicHunter bayesian multimedia retrieval system. In *Advanced Digital Libraries ADL'96 Forum*, Washington, DC, May 1996.

[3] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, third edition, 2001.

[4] P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *Proc. IJCNN-90, International Joint Conference on Neural Networks, Washington, DC*, volume II, pages 279–285, Piscataway, NJ, 1990. IEEE Service Center.

[5] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja. Self-organizing maps as a relevance feedback technique in content-based image retrieval. *Pattern Analysis & Applications*, June 2001. In press.

[6] J. T. Laaksonen, J. M. Koskela, S. P. Laakso, and E. Oja. PicSOM - Content-based image retrieval with self-organizing maps. *Pattern Recognition Letters*, 21(13-14):1199–1207, December 2000.

[7] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28(1):84–95, Jan. 1980.

[8] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. Computer Science Series. McGraw-Hill, 1983.

TOWARDS REAL-TIME SHOT DETECTION IN THE MPEG-COMPRESSED DOMAIN

Janko Calic and Ebroul Izquierdo
Multimedia and Vision Research Lab
Dept. of Electronic Engineering, Queen Mary, University of London
e-mail: {janko.calic, ebroul.izquierdo}@elec.qmw.ac.uk
Mile End Road, E1 4NS London, United Kingdom
Tel: +44 20 78825530, Fax: +44 20 7882 7997

ABSTRACT

As content based video indexing and retrieval has its foundations in the prime video structures, such as a shot or a scene, the algorithms for video partitioning have become crucial in contemporary development of digital video technology. Conventional algorithms for video partitioning mainly focus on the analysis of compressed video features, since the information relevant to the partitioning process can be extracted directly from the MPEG compressed stream and used for the detection of shot boundaries. However, most of the proposed algorithms do not show real time capabilities that are essential for video applications. This paper introduces a real time algorithm for cut detection. It analyses the statistics of the features extracted from the MPEG compressed stream, such as the macroblock type, and extends the same metrics to algorithms for gradual change detection. Our analysis led to a fast and robust algorithm for cut detection. Future research will be directed towards the use of the same concept for improving the real-time gradual change detection algorithms. Results of computer simulations are reported.

1. INTRODUCTION

The contemporary development of various multimedia compression standards combined with a significant increase in desktop computer performance, and a decrease in the cost of storage media, has led to the widespread exchange of multimedia information. The availability of cost effective means for obtaining digital video has led to the easy storage of digital video data which can be widely distributed over networks or storage media as CDROM or DVD. Unfortunately, these collections are often not catalogued and are accessible only by sequential scanning of the sequences. To make the use of large video databases feasible, we need to be able to automatically index, search and retrieve relevant material.

It is important to stress that even with leading edge hardware accelerators, factors such as algorithm speed and storage costs are concerns that must still be addressed. For example, although compression provides tremendous space savings, it can often introduce processing inefficiencies when decompression is required to perform indexing and retrieval. With this in mind, one of the main considerations in our development of a system for video retrieval is initially an attempt to enhance access capabilities within existing compression representations.

Since the identification of the temporal structures of video is an essential task of video indexing and retrieval [1], shot detection has been generally accepted to be an essential and first step in indexing algorithm implementation. We define a *shot* as a sequence of frames that were (or appear to be) "continuously captured from the same camera" [1]. A *scene* is defined as a "collection of one or more adjoining shots that focus on an object or objects of interest" [3].

Shot change detection algorithms can be classified according to the features used for processing into uncompressed and compressed domain algorithms. Algorithms in the uncompressed domain utilise information directly from the spatial video domain: pixel-wise difference [4], histograms [5], edge tracking [6], etc. These techniques are computationally demanding and time consuming, and thus inferior to the compressed features based approach.

Since compressed-domain methods have become dominant in this field [1] we will concentrate more on features extracted directly from compressed video. We will specifically focus on the use of MPEG compressed streams.

In this context, Yeo and Liu [7] proposed an initial method for compressed domain shot detection using a sequence of reduced images extracted from DC coefficients in the transformation domain called the DC sequence. An interesting approach was proposed by Lee *et al*. [8], where they exploit information from the first few AC coefficients in the transformation domain, and track binary edge maps to segment the video. Two approaches similar to our method are proposed by Kobla et al. [9] and Pei et al. [10], where the authors have analysed information extracted from MPEG motion estimation variables in various ways.

In this paper, the main goal is to develop a new approach to the fundamental problems found in a system for real-time video retrieval, searching and browsing. The initial research objectives are directed towards the performance of the main video processing algorithms in the compressed domain, using the established international video standards: MPEG1-2, H.263 and in future MPEG4. Further steps will be concerned with a novel approach to the specification of low level image features based on the scale space paradigm that will be used for the definition of image descriptors, colour clustering and shape based image retrieval. This approach should introduce improvements in video retrieval with low access latency, as well as advances in processing speed and algorithm complexity.

The proposed algorithm shows the highest accuracy in detection of abrupt changes, while the few attempts to implement fast detection of gradual changes show some very interesting results: motion features extracted from the MPEG stream showed high instability, while the referencing measure introduced in Section 2 is vague for the detection of longer transitions.

This paper is organized as follows. In Section 2 we present the algorithm for detection of abrupt shot changes. Section 3 describes the gradual transition detection algorithms continuing with the same approach, as well as adding some interesting conclusions. Overall results are presented in Sections 6, while finally, in Section 7, we give conclusions and a summary of the paper.

2. SCENE CHANGE DETECTION

MPEG-2 encoders compress video by dividing each frame into blocks of size 16x16 called *MacroBlocks* (MB) [11]. A MB contains information about the type of temporal prediction and corresponding vectors used for motion compensation. The character of the MB prediction is defined in a MPEG variable called *MBType*. It can be: *Intra* coded, *Forward* referenced, *Backward* referenced or *Interpolated*. Clearly, a MPEG stream has a high temporal redundancy within a shot. Thus, a continuously strong inter-frame reference will be present in the stream as long as no significant changes occur in the scene. The "amount" of inter-frame reference in each frame and its temporal changes can be used to define a metric, which measures the probability of scene change in a given frame. We propose to extract MBType information from the MPEG stream and to use it to measure the "amount" of inter-frame reference. Scene changes are then detected by thresholding the resulting function.

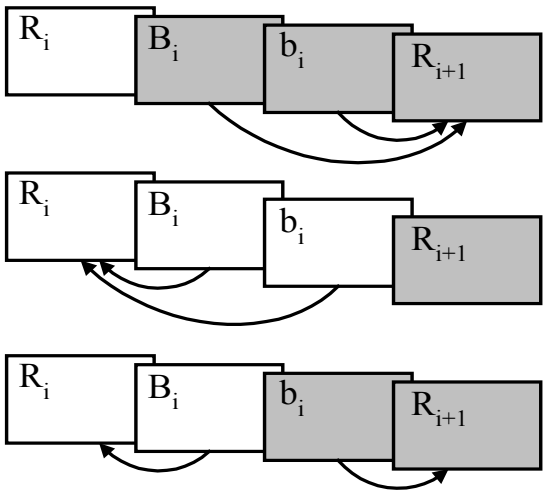


Figure 1 Possible positions of the cut in a frame triple

A *cut* is defined as an abrupt scene change between two consecutive frames. Without loss of generality we assume that a MPEG *Group Of Pictures* (GOP) will have the structure [IBBPBBPBBPBBPBB]. Observe that this structure can be split into groups of three frames having the form IBB or PBB (a triplet).
In the sequel I or P reference frames are denoted as R_i^j, where j=0,1,…N, i=0,1,…,N/3, and N is number of frames in the sequence. The first B referenced frame in the triplet is denoted as B_i^j while the second one is denoted b_i^j. Thus, any GOP can be expressed by frame-triplets of the form: $R_i^{j=3i-1} B_i^{j=3i} b_i^{j=3i+1}$. This convention can be easily generalized to any other GOP structure.

The location of a possible cut in a triplet is depicted in Fig. 1. If the referenced frame B_i^j is the first frame in the next shot (shaded frames), there will be a significant

percentage of inter-frame referenced MBs from both B_i^j and b_i^{j+1} in the next reference frame R_{i+1}^{j+2}. If the scene change occurs at R_{i+1}^j, then the previous frames B_i^{j-2} and b_i^{j-1} will be mainly referenced to R_i^{j-3}. Finally, if the scene change occurs at b_i^j, then B_i^{j-1} will be referenced to R_{i+1}^j while b_i^j will be referenced to R_{i+1}^{j+1}.

If two frames are strongly referenced then most of the MBs in each frame will have the corresponding type, forward, backward or interpolated, depending on the type of reference. Thus, we can define a metric for the visual frame difference by analyzing the percentage (or simply the number) of MBs in a frame that are forward referenced and/or backward referenced.

Let $\Phi_T(j)$ be the set containing all forward referenced MBs and $\mathrm{B}_T(j)$ the set containing all backward referenced MBs in a given frame with index ***j*** and type **T**. Then we denote the cardinality of $\Phi_T(j)$ as $\text{☞}_T(j)$ and the cardinality of $\mathrm{B}_T(j)$ as $\text{👌}_T(j)$. The metric $\Delta(j)$ used to determine a cut is defined as:

$$\Delta(j)=\begin{cases} \text{👌}_B(j)+\text{👌}_b(j+1), & \text{if } j^{th} \text{ frame is a B frame} \\ \text{☞}_B(j-2)+\text{☞}_b(j-1), & \text{if } j^{th} \text{ frame is a R frame} \\ \text{☞}_B(j-1)+\text{👌}_b(j), & \text{if } j^{th} \text{ frame is a b frame} \end{cases}$$

Cut positions are determined by thresholding using either predefined constant threshold or an adaptive one.

3. GRADUAL CHANGES DETECTION

The next step in the implementation of a shot changes detection algorithm is the detection of gradual changes. Gradual transitions do not show such significant changes in any of the features, and thus are more difficult to detect. Due to advances in digital video editing, there are various types of gradual changes: *dissolves*, where the first shot frames become dimmer, while the second ones become brighter and are superimposed on the first shot frames; *wipes*, where the image of the second shot replaces the first one in a regular pattern, such as vertical line, etc. Since there is inevitably additional processing in feature analysis for gradual changes extraction, real time implementation is even farther from reality than it is for basic cut detection. Because of this, the main efforts are directed towards an improvement of the solutions for gradual changes detection.

Given that the initial approach was to use information incorporated in the process of motion estimation and temporal prediction, the first feature to be analysed was the set of *Motion Vectors* (MV) from the MPEG stream. The extracted set of vectors is a three-dimensional vector field, and within it there are numerous features that could be analysed for changes detection, such as statistical distribution of vector intensities and angles, gradients, divergences, etc. Unfortunately, the results show something else. Theoretically, the set of MV should show very typical behaviour during gradual transitions. In reality there is a decrease in the amount of defined MV per frame in transition regions due to an increase in the number of *Intra* coded MBs, so that the process of MV analysis becomes highly unstable. This problem becomes less important in MPEG streams with higher bit rates, but is never avoided entirely. Fortunately, there is some important information that can be extracted using MV like camera movement, panning, zooming, object appearance and disappearance, etc. However, these video features belong to processes on a higher level, such as video abstraction, scene analyses, etc. We want to remain at the lowest level of video partitioning.

If one wonders if it is possible to use the significant instability of MV information as a sign of the shot changes, we must draw attention to the fact that the information about MV definition is actually information about MBType. Moreover, since the approach of the abrupt shot change detection algorithm was based on

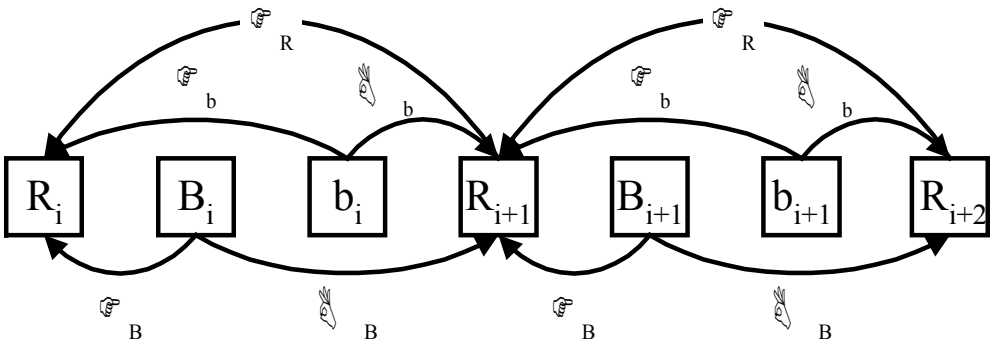


Figure 2 Structure of two frame triples

analyses of the inter-frame references, it would be natural to apply the same paradigm to an algorithm for the detection of gradual transitions.

Obviously, the metrics and the analyses have to be different from the ones used in cut detection. The process of gradual changes "calculate a frame-to-frame distance and then perform some kind of tracking over it" [12], so that we need difference metrics between two frames within random distance. The inter-frame reference from Section 2 can be used as inverted difference metrics, but must be generalised to random distance and random frame type for this purpose.

Let us analyse Figure 2, which shows the general frame structure in two frame-triples *i* and *i+1*.

Since it is important to define an inter-frame reference for each frame and for each frame distance, there will be five different types of local inter-frame references $\delta_{X_i X_{i+1}}(i)$ within a frame triple that will form overall inter-frame reference $1/\Delta_D(i)$ at random distance $\boldsymbol{D}$. The definition of local inter-frame reference is given as follows:

$$\delta_{R_i^j R_{i+1}^{j+3}}(j) = \max(\ \text{☞}_B(j+1)\cdot \text{✍}_B(j+1),\ \text{☞}_b(j+2)\cdot \text{✍}_b(j+2),\ \text{☞}_R(j+3)\ )$$

$$\delta_{R_i B_{i+1}}(j) = \max(\text{☞}_B(j+1), \delta_{R_{i+1}R_{i+2}}(j)\cdot \text{✍}_B(j+1))$$
$$\delta_{R_i b_{i+1}}(j) = \max(\text{☞}_b(j+2), \delta_{R_{i+1}R_{i+2}}(j)\cdot \text{✍}_b(j+2))$$

$$\delta_{B_i R_{i+1}}(j) = \max(\text{✍}_B(j), \delta_{R_i R_{i+1}}(j-1)\cdot \text{☞}_B(j))$$
$$\delta_{b_i R_{i+1}}(j) = \max(\text{✍}_b(j), \delta_{R_i R_{i+1}}(j-2)\cdot \text{☞}_b(j))$$

To calculate the overall inter-frame reference, the local values are simply multiplied, with the frame types and positions in mind:

$$\frac{1}{\Delta_D(i)} = \overbrace{\delta_{X_i R_{i+1}}}^{\exists\delta} \cdot \prod_{i+1}^{i+D-1} \delta_{R_i R_{i+1}} \cdot \overbrace{\delta_{R_{i+D-1} X_{i+D}}}^{\exists\delta}$$

Detection of changes is implemented by applying the twin comparison algorithm proposed by Zhang *et al.* [13] to the inter-frame difference $\Delta_D(i)$ as the algorithm metrics.

4. RESULTS

The collection of C++ classes called Mpeg Development Classes, implemented by Dongge et al. [14], was used as the main tool for manipulating the MPEG streams, while Berkeley mpeg2codec was used as the reference MPEG codec. Test sequences were produced by Multimedia & Vision Research Lab, Queen Mary, University of London.

To evaluate the behaviour of first algorithm, we generated sample MPEG2 video sequence *shot.m2v* with three abrupt shot changes at 6th, 16th and 23rd frame. As depicted in Figure 3, the first cut is positioned at rear b frame, and as proposed, it is clear that the level of forward reference is high at previous B frame ☞(5), and that at the present frame there is strong backward referencing ✍(6). In the same way, for the 16th I type frame there are significant ☞(13) and ☞(14), and the 23rd B type frame has strong ✍(23) and ✍(24). This procedure showed excellent results for abrupt changes with almost 100% accuracy.

Evaluation of gradual changes detection is always a delicate issue due to the variety of potential transition types and digital editing tricks. The dataset used for evaluation cannot be considered as a high reference dataset, but it has shown some significant results in this case.

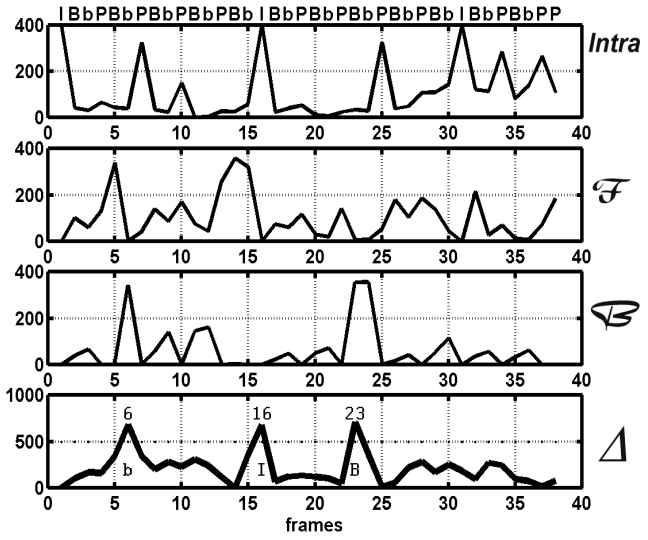


Figure 3 Detection of the cuts on the 6th, 16th and 23rd frame in a sequence Janko.m2v

The evaluation of the algorithm that analyses motion vector fields turned out to be very poor, since we obtained less than 5% of MB with defined motion vectors in frame areas neighbouring the abrupt shot change.

Considering the method that uses inter-frame reference metrics, let us analyse Figure 4, which shows $\Delta_D(i)$ defined in Section 3.

There are three shot changes: cut on the 48th frame, wipe from the 82nd to the 121st frame and dissolve from the 160th to the 183rd frame. The graph shows clear detection of all three changes replaced by 20 frames because of the

comparison window. However, this method showed weaknesses on sequences with high motion during the shot changes.

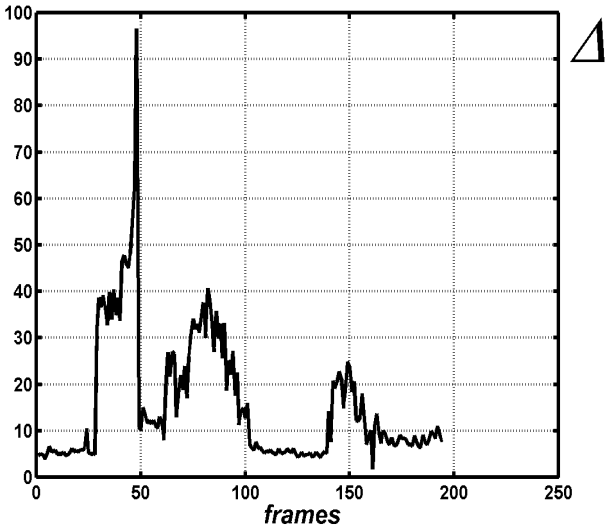


Figure 4 Difference metrics Δ for three types of gradual changes: cut, wipe and dissolve

5. CONCLUSIONS

In this paper, we propos a new shot change detection algorithm based on the motion information extracted from the MPEG video stream. First, we introduced a method for abrupt changes detection that uses inter-frame reference derived only from the statistics of the *MacroBlock* types. Second, we applied the similar inter-frame reference metrics in the algorithm for gradual shot detection, where we compared the frame difference within a random distance using the twin comparison algorithm. Finally, we introduced experimental results in Section 4. The results show high accuracy in abrupt shot detection. However, the method for gradual changes detection shows sensitivity to high motion during the shot changes, and thus needs refinement

We are investigating the possibilities of improving the real time gradual shot changes detection using multidimensional clustering of the MPEG compressed features. Additional MPEG features should make the proposed algorithm less sensitive to strong motion during the shot changes without losing the real time capabilities.

6. REFERENCES

[1] Zhang H. J., "Content-based Video Browsing and Retrieval", from "Handbook of Multimedia Computing" editor-in-chief Furht B., CRC Press, Boca Raton, Florida, USA, 1999.

[2] Gargi U., Strayer S., "Performance Characterisation of Video-Shot-Change Detection Methods", IEEE Trans. on Circuits and Systems for Video Technology, Vol.10, No.1, February 2000

[3] J.S. Boreczky and L.A. Rowe, "A Comparison of Video Shot Boundary Detection Techniques", Storage & Retrieval for Image and Video Databases IV, I.K. Sethi, and R.C. Jain, Editors, Proceedings SPIE 2670, pp. 170-179, 1996.

[4] H. Zhang, A. Kankanhalli and W. Smoliar, "Automatic partitioning of full-motion video", Multimedia Systems, vol.1, no.1, pp. 10-28, 1993.

[5] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances", Proc. IFIP 2nd Working Conf. Visual Database Systems, pp.113-127, 1992.

[6] R. Zabih, J. Miller and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks", Proc. ACM Multimedia '95, pp.189-200, 1995.

[7] Boon-Lock Yeo, Bede Liu, "Rapid scene analysis on compressed video", IEEE Transactions on Circuits & Systems for Video Technology, vol.5, no.6, Dec. 1995, pp.533-44. Publisher: IEEE, USA

[8] Seong-Whan Lee, Young-Min Kim, Sung Woo Choi, "Fast scene change detection using direct feature extraction from MPEG compressed videos", IEEE Transactions on Multimedia, vol.2, no.4, Dec. 2000, pp.240-54. Publisher: IEEE, USA

[9] Kobla V., Doermann D.S., Lin K. -I., Faloutsos C., "Compressed domain video indexing techniques using DCT and motion vector information in MPEG video", Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases V, Volume 3022, pp. 200-211, February 1997.

[10] Soo-Chang Pei, Yu-Zuong Chou, "Efficient MPEG compressed video analysis using macroblock type information", IEEE Transactions on Multimedia, vol.1, no.4, Dec. 1999, pp.321-33. Publisher: IEEE, USA

[11] LeGall D., Mitchell J.L., Pennbaker W. B., Fogg C.E., "MPEG video compression standard", Chapman & Hall, New York, USA, 1996

[12] Bescos J., Martinez J.M., Cabrera J., Menendez J.M., Cisneros G., "Gradual shot transition detection based on multidimensional clustering", 4th IEEE Southwest Symposium on Image Analysis and Interpretation. IEEE Comput. Soc. 2000, pp.53-7. Los Alamitos, CA, USA

[13] H. Zhang, A. Kankanhalli and S. Smoliar, "Automatic partitioning of full-motion video", Multimedia Systems, vol. 1, pp. 10-28, July 1993.

[14] Dongge Li, Sethi I. K. "MDC: a software tool for developing MPEG applications", Proceedings IEEE International Conference on Multimedia Computing and Systems. IEEE Comput. Soc. Part vol.1, 1999, pp.445-50 vol.1. Los Alamitos, CA, USA

TRACKING OF OBJECTS IN VIDEO SCENES WITH TIME VARYING CONTENT

Amal Mahboubi, Jenny Benois-Pineau, Dominique Barba
IRCCyN UMR n° 6597 CNRS
EPUN, rue Christian Pauc La chantrerie BP 60601 44306 NANTES France
Tel: +33 2.40.68.30.46 Fax: +33 2.40.68.32.32
e-mail: amahboub@ireste.fr jbenois@ireste.fr dbarba@ireste.fr

ABSTRACT

This paper proposes a method for tracking of objects contained in video sequences. Each video object is represented by a set of polygonal regions. The tracking of this model along moving sequence is based on a detecting and indexing new objects in a video scene.

1 INTRODUCTION

The new ongoing standard of video representation and coding MPEG4 [1] gives tremendous possibilities for the composition of heterogeneous video scenes combining video objects of various nature. The main challenge behind MPEG4 technology is the development of efficient and truly automatic methods for extracting and tracking of objects in video. Once video objects are known at each time in a video, they can be manipulated, put into another scene etc…. Numerous research works have been developed recently [2,3] devoted to the problem of automatic tracking of a selected video object (VOP) in a scene. In this paper we address the problem of tracking in case of changing content, with strong structural changes in a known object and new objects witch appear in the scene.
The method with the extraction of video object from a complex natural video scene at the initial time instant, using a fine spatial partition of image plane. The geometry of each spatial primitive is represented by a piece-wise linear approximation of the border. Affine motion model of each polygonal region is estimated by means of gradient descent method. Then all regions are classified semantically by means of human interaction. Then connected regions are merged in each semantic class to build a hierarchical representation of the scene using motion homogeneous criteria. The tracking of changed content is based on motion estimation of regions along the time and on textural and topological coherence measures.
The paper is organized as follows: section 2 describes the initialization of object-based partition of video scenes. General tracking scheme is described in section 3. Section 4 represents the indexing of VOPs in case of time-varying content. Finally main results of the tracking are presented in section 5.

2 INITIALISATION OF OBJECT-BASED PARTITION OF VIDEO SCENES

To extract objects to be tracked, a spatial color–based segmentation of video frame at the initial time instant is applied. The spatial segmentation (see Figure 3-c) is the result of a modified color based watershed algorithm we developed in [4]. For each spatial region a polygonal representation is constructed using a piece-wise linear approximation of its border. To build the spatio-temporal structure we estimate the motion of each polygonal region by the gradient descent method. The motion vector parameters Θ correspond to a reduced affine 2D motion model $\Theta = (t_x, t_y, k, \theta)^T$. According to it, an elementary displacement vector $(dx,dy)^T$ at each pixel position (x,y) in a given region is expressed as:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} tx \\ ty \end{bmatrix} + \begin{bmatrix} div & -rot \\ rot & div \end{bmatrix} \begin{bmatrix} x - x_g \\ y - y_g \end{bmatrix}.$$

Here x_g,y_g are the coordinates of the gravity center of the region tx,ty, are translation parameters, div is a zoom parameter and rot is a rotation parameter [5].
Theses regions should be labeled semantically to provide VOPs corresponding to meaningful objects in a scene. Purely automatic labeling is possible only for simple scenes , where a strong difference of dynamic and textural characteristics of objects and the background is observed. In general case of natural scenes, an object can be partly static an thus can not be distinguished from the background based on motion. The color and texture of object can be similar to the background. Therefore, a user interaction is required to completely extract objects in a general case. We propose a minimal human intervention.

The user creates a binary semantic class mask on the first frame by encircling objects, here we have an image with 0 in the background and 1 inside objects (the encircled area). This binary image called "user mask" is then used for the initial VOP labeling.

Each polygonal region is superimposed on the user mask to get the initial classification. Thus the object threshold noted Th_{obj} and three semantic classes can be introduced:

1. "Object": is the class for objects in the scene.
2. "Background": this class denotes generally the scene background.
3. "Uncertain": this class represents the ambiguous area on VOPs borders.

The semantic label of each region is obtained according to the following.

$$Label(R_i) = Arg\ Max(Form(R_i), Back(R_i), Unce(R_i))$$

$$Form(R_i) = \begin{cases} 1 & if\ Card(\Omega R_i) > Th_{obj} \\ 0 & \text{otherwise} \end{cases}$$

$$Back(R_i) = \begin{cases} 1 & if\ Card(\Gamma R_i) < 1 - Th_{obj} \\ 0 & \text{otherwise} \end{cases}$$

$$Unce(R_i) = \begin{cases} 1 & if\ 1 - Th_{obj} < Card(\Omega R_i) < Th_{obj} \\ 0 & \text{otherwise} \end{cases}$$

where :

$$\Omega R_i = \{p_i \in R_i / Mask[p_i] = 1\}$$

$$\Gamma R_i = \{p_i \in R_i / Mask[p_i] = 0\}$$

The labeled spatial regions constitute a fine partition of the image plane which is too redundant with regard to the scene content (see Figure 3-b). Therefore, a motion-based merging process is necessary to construct more meaningful region-based partition. We follow the merging strategy proposed in [6] to construct a nested hierarchical polygonal partition inside each semantic class see Figure 4.

Finally, each VOP is indexed in the video scene by the following method.

Each polygonal region in the image plane corresponds to a region-node in the region adjacency graph (RAG). Starting from an arbitrary "Object Class" node in RAG, all the graph is traversed by "In-Depth Search" algorithm and the maximal sub-graph with only Object Class nodes is isolated. This sub-graph corresponds to a connected VOP in the scene. All region-nodes receive a label we call "Object Index". The process is re-iterated for all remaining "Object class" nodes with incremented "Object Index" label.

Resulting form this process, the label of each region in image plane partition is set to "Uncertain", "Background" or its own "Object Index" value corresponding to the VOP index.

3 TRACKING SCHEME

The principle that guides our tracking scheme was developed in [5] for polygonal partition of video frames Based on affine motion model of 2D apparent motion, it consists in projecting of polygons in time axis direction, adjusting of predicted borders by an active contour model, segmenting of regions with changed content and merging of regions at time *t+1*. Thus the spatio-temporal partition S^{t+1} is obtained from S^t. In the scenes with changing content, it is necessary to label new regions as belonging to new or pre-existing VOPs, to the background or to Uncertain class. The method presented here incorporates solutions for labeling the new regions.

A new region is the result of prediction and adjustment steps of tracking scheme. When projecting a spatio-temporal partition S^t to the next frame with affine model, overlapped and uncovered areas are formed in image plane at time *t+1*. In our previous work [5] we studied in detail processing of occlusions in overlapped areas. For these occlusions their motion–based assignment to already existing regions was proposed. In this work we concentrate to a more difficult type of occlusions without "pre-history", that is to *uncovered* regions. They can appear in the neighborhood of VOPs, on the borders of video frames in case of background motion, inside VOPs (self-uncovered areas). The second problem is to correctly label the "*cut* regions" which are issued from the motion-based segmentation of projected regions with increased motion compensation error. The *uncovered* and *cut* regions contain both parts of new objects and of the background or pre-existing objects. Thus the problem here is to correctly label these regions with "Object Index" value, "Background" or "Uncertain" labels. We show in the next section how this goal can be achieved.

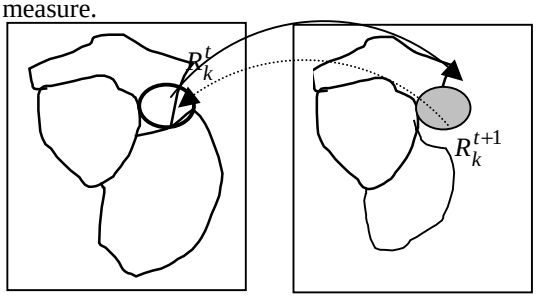
4 NEW REGION LABELING

New region having different origins, we establish different rules for each type: the *cut* region labeling is based on motion estimation, the *uncovered* region labeling is based on the texture information and topological analysis of its spatio-temporal neighborhood.

4.1 Labeling of uncovered regions

Regions in uncovered areas after projection of segmentation can be adjacent to VOPs borders, or to be situated inside an articulated VOP. To label these areas two measures are combined: a score of pixels belonging to a specific class (Object, Uncertain, Background) in the past reference frame on one hand and a texture similarity measure in the current frame on the other hand. These two measures are mixed in one decision rule.

The first measure denoted *Score* refers to the class to which each pixel of region R^{t+1} back-projected into frame I^t does belong. The second measure denoted L indicates the class of a region in the neighborhood of R^{t+1} in the current frame, which has the most similar texture to the texture of R^{t+1}. A trust weights are assigned to each of these measures and the resulting class label for the region R^{t+1} is that maximizing the global trust measure.



t t+1

Figure 1. Diagram of a back-projection

Let see in Figure 1 the new region $R_k{}^{t+1}$ in grey and its back-projection $R_k{}^t$ into the frame I^t. First we compute its score as:

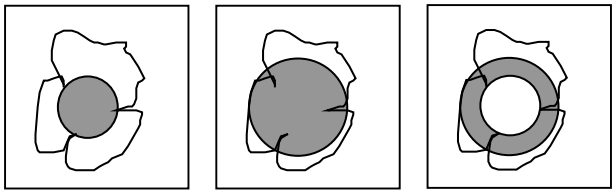
$$Score(C_i / R_k^{t+1\to t}) = \sum_{(x,y)\in R_k^{t+1\to t}} \delta(o(x,y)-\omega_i)$$

where $o(x,y)$ is the observation - label

$\Omega = \{\omega_i, i = 1..3\}$ the class label corresponding to Object, Background, Uncertain

$$\delta(a-b) = \begin{cases} 1 & if \quad a = b \\ 0 & \text{otherwise} \end{cases}$$

$$Weight_{S_i} = Score(C_i / R_k^{t+1\to t}) \Big/ Card(R_k^{t+1\to t})$$



-a- -b- -c-

Figure 2. Diagram of uncovered region and his neighborhood

The computation of texture similarity measure is based on assumption of gaussian grey-level distributions in limited windows surrounding the given region. This windows are constructed by dilating the region R_k. Figure 2 depicts the method: in figure 2-a the region is denoted by hatched pattern, figure 2-b presents a dilated region, the resulting windows are shown at figure 2-c (hatched pattern).
Thus the parameters of windows in neighborhood $\{R_{ki}\}$ will be mean μ_{ki} and variance σ_{ki}^2 .
The neighbor likelihood is computed as [7]:

$$L_k = \max_i \tau_{ki} \ , \ \tau_{ki} = (x-\mu_{k_i})^2 \Big/ 2\sigma_{ki}^2 \quad ; i = 1..N$$

where N is the regions number in the neighborhood of R_k .
We define $M_{L_{kj}}$ as the neighborhood average of the region R_k for the class j and $\overline{M}_{L_{kj}}$ as his class complementary:

$$\mathrm{M}_{\mathrm{L_{kj}}} = 1/h \sum_{i=1}^{h} \tau_{kji}, \quad \text{with} \quad h \le N, \ i = 1...h \ \text{ all the regions in the class j}$$

$$\overline{M}_{Lkj} = 1/\bar{h} \sum_{i}^{\bar{h}} \tau_{kci} \quad \bar{h} \le N, \ h + \bar{h} = N, \ c = 1..3, \ j = 1..3, c \ne j$$

the trust weight of the likelihood characterizes its relative deviation from the mean likelihood:

$$Weight_{L_{kj}} = (M_{L_{kj}} - \overline{M}_{L_{kj}}) \Big/ \max(M_{L_{kj}}, \overline{M}_{L_{kj}}) \quad ; j = 1..3 \quad \text{the classes}$$

Finally the class label to the region is given by:

$$Label(R_k) = ArgMax(Weight_{S_{kj}} + Weight_{L_{kj}}); \ j = 1..3$$

4.2 Labeling of cut regions

When a region is re-segmented at time *t+1*, the resulting set of regions could contain parts of new objects and parts of existing already labeled regions at time *t*. The problem here is to define which sub-region corresponds to a new moving object superimposed on the pre-existed background or to a new detail in the pre-existing object. The method we propose is based on the measurement of a differential motion activity of each sub-region. The assumption here is that a new significant region belonging to a new objets strongly changes its motion between two successive frames. Let R_{ki}^{t+1} denote a sub-region resulting from motion–based segmentation of region R_k at time *t+1*. Let θ_k^t be the motion parameter vector of R_k at time *t*, θ_{ki}^{t+1} is the motion vector of R_{ki}^{t+1}, If the region R_{ki}^{t+1} is back-projected into the image plane at time *t*, then to the pixel position (*x,y*) at *t+1*corresponds the position (*x+dx,*

$y+dy$). The elementary displacement vectors $\vec{d}\left(x,y,\theta_{ki}^{t+1}\right)$, $\vec{d}\left(x+dx^{t+1},y+dy^{t+1},\theta_k^t\right)$ are computed at time t+1 and t for each pixel position (x,y) and (x+dx, y+dy) respectively with motion parameters of the regions R_{ki}^{t+1} and R_k^t.

Then the measure of differential motion activity we introduce will be expressed as :

$$\Delta_{mvt}\left(R_{ki}^{t+1}\right)=\frac{1}{Card R_{ki}^{t+1}}\sum_{(x,y)\in R_{ki}^{t+1}}\left\|\vec{d}\left(x,y,\theta_{ki}^{t+1}\right)-\vec{d}\left(x+dx^{t+1},y+dy^{t+1},\theta_k^t\right)\right\|^2$$

If this measure is stronger than a threshold, then the region R_{ki}^{t+1} is labeled as Object-class region.

5 RESULTS AND PERSPECTIVES

The proposed automatic indexing of Objects in scenes with a changed content was experimented on the sequence "Children" in CIF format at 12 ips (MPEG4 test sequences). Some results are given in the table 1.

Here we compare the performance of our Automatic labeling to the visual tracking result considered in frames at time t=3 and t=5 .

Frame	Type	Region-Number	Correct-classification
3	Cut	7	5
	Uncovered	28	20
5	Cut	8	4
	Uncovered	34	25

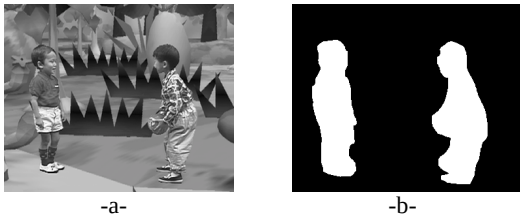
Table 1. Statistic performance

The result of semantic classification uncovered areas are shown at Figure 5. Here the left image depicts these areas in grey, the right image correspond to the results of classification.

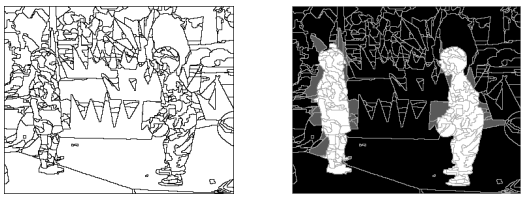
The labeling of "*cut* regions" is shown in Figure 6. It can be seen that new moving objects are correctly labeled.

Finally, the tracking results are shown in Figure 7.

These first results are promising. They show that in the sequence with a changed content and a strong relative motion of objects with the background the main objects are detected successfully. Nevertheless the tracking of the semantic classification can present some errors in thin areas.



-a- -b-



-c- -d-

Figure 3. The process of the first semantic classification. -Sequence "Children" frame at t=3-
a) original frame b) user mask, c) spatial segmentation, d) result of the semantic classification.

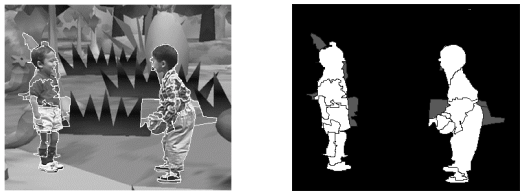
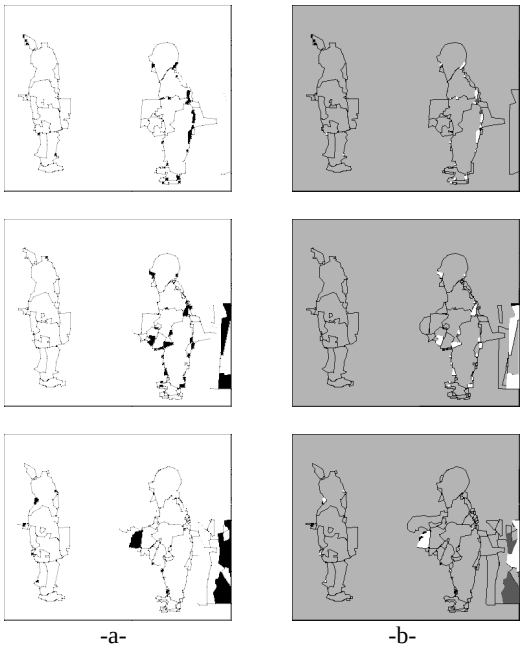


Figure 4. The result of the manual semantic classification for the first frame (black for Background, white for the VOPs and grey for Uncertain)



-a- -b-

Figure 5. The result of the semantic classification of the uncovered regions
a) the new uncovered regions in black, b) the black is for the "Background", the white for the "Object", the dark grey is for the "Uncertain". The pale grey is used only to display all regions.

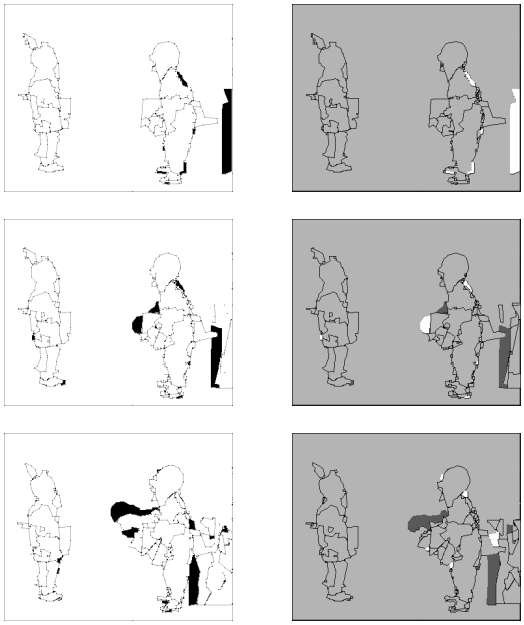
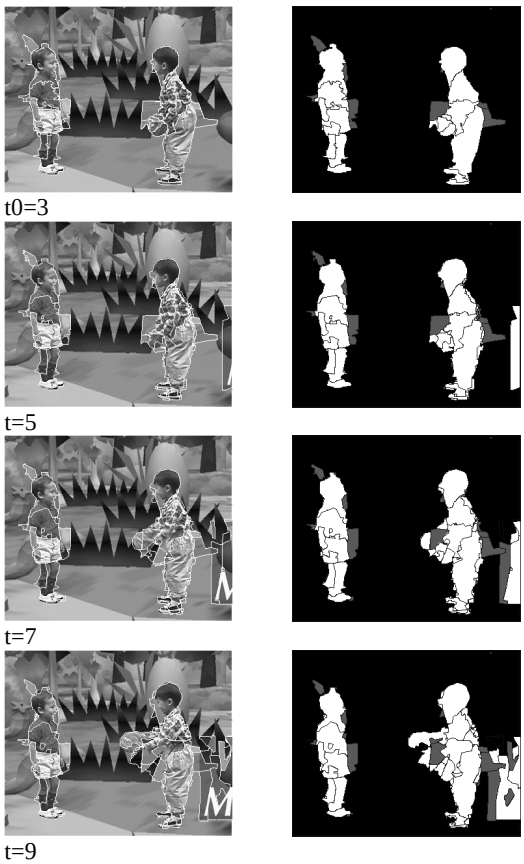


Figure 6. The result of the semantic classification of "cut regions"

a) the new *cut* regions in black, b)the black is for the "Background", the white for the "Object", the dark grey is for the "Uncertain". The pale grey is only the color of the support frame



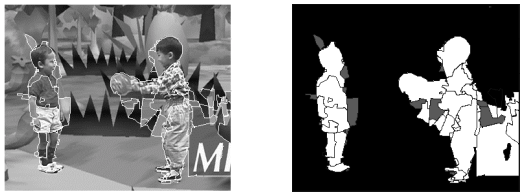


Figure 7. The result of the polygonal tracking
a) the original frames with the segmentation, b) the semantic frames: the black is for the "Background", the white for the "Object", the grey is for the "Uncertain".

This work is supported by the project RNRT OSIAM.

6 REFERENCES

[1] ISO/IEC JTC1/SC29/WG11 N2202. Information technology-Coding of audio-visual objects: Visual. ISO/IEC 14496-2 Committee Draft (MPEG4: Visual). Tokyo, March 1998.

[2] F. Marques, C. Molina. « Object tracking for content–based functionalities », Proc. VCIP'97, San-José, CA (USA), Feb. 1997, pp. 190-199

[3] S. Jehan, M. Barlaud, G. Aubert « Detection and tracking of moving objects using a new level set based method », ICPR'2000, Barcelone, Septembre 2000

[4] A. Mahboubi, J. Benois-Pineau, D. Barba "Segmentation spatiale couleur des images par une approche morphologique et hierarchique", CORESA, Poitiers, France, session -algorithmes pour fonctionnalités avancées-, 19-20 Oct 2000.

[5] L. Wu, J. Benois-Pineau, D. Barba, "Spatio-temporal segmentation of image sequences for object oriented low bit-rate image coding", Signal Processing:Image Communication, a EURASIP journal, vol 8(6), pp 513 – 544, sept 1996.

[6] J. Benois-Pineau, F. Morier, D. Barba, H. Sanson "Hierarchical segmentation of video sequences for content manipulation and adaptive coding", Signal Processing 66 pp. 181-201, 1998.

[7] S. M. Kay, " fundamentals of statistical signal processing" Prentice Hall International, Inc, USA, 1993

PSYCHOLOGICALLY RELEVANT FEATURES OF COLOR PATTERNS

Jan Restat
Abteilung Interaktive Medien, Heinrich-Hertz-Institut für Nahcrichtentechnik,
Einsteinufer 37, 10587 Berlin, GERMANY
Tel. +49 30 31002 779 e-mail: restat@hhi.de

ABSTRACT

In an empirical study, 36 subjects rated the similarity of 20 color patterns. The analysis of the similarity ratings with hierarchical clustering and factor analysis yielded four relevant basic categories of pattern features. These categories turned out to be 1. directionality, 2. color purity, 3. regularity and complexity and 4. (somewhat surprising) purpleness. These categories are in principle similar to an earlier study, but partly the results differ. Taken together, a relative stable core of general categories for pattern similarity ratings is emerging, which play an important role for most subjects and patterns; as well as other features, which may play a role only for certain pattern combinations or few subjects.

1 INTRODUCTION

Today's image retrieval systems frequently utilize the "query by example" input method. The user specifies a picture from a random sample which is "most similar" to his or her target image. While this "query by example" method is very comfortable for input, its satisfying *output* depends on the successful definition of "similarity": it is crucial to know which of all possible features of an example image the human user expects to be similar in the returned images from the database. In the following, we report some empirical findings on the psychological dimensions of colored texture and pattern similarity.

Besides color and shape, texture is one of the basic dimensions of images. While there has been some research on the psychological dimensions of gray textured patterns [1], up to now, there has been reported only one study on the categories of human similarity ratings of color patterns (Mojsilović, Kovačević, Kall, Safranek & Ganapathy, 2000 [2]). Using hierarchical clustering and multidimensional scaling techniques, Moijsilović et al. determined the basic categories in judging similarity of color patterns to be 1. overall color (one dominant color vs. several colors), 2. directionality and orientation, 3. regularity and placement, 4. color purity (pale vs. saturated colors) and 5. complexity and heaviness.

To test the validity of these findings, we repeated the empirical study, supplementing the basic design with new texture material, a new computer aided stimulus presentation, and a higher number of participants as well as a higher number of ratings per participant. In the following section, the methodology of the empirical setting is described. In section three, the results are presented, and in the final section, conclusions are discussed.

2 METHOD

2.1 Selection of Stimuli

Taken from several pattern databases, 106 patterns of great variety were presented in a short pre-study to 12 participants which had to choose 10 especially distinctive patterns and 10 patterns which were similar to several other patterns. The combination of these ratings were used to identify twenty patterns which included very distinctive as well as "middle-of-the-road"-patterns. These 20 patterns (see Fig.1) were then used in the actual study[1].

2.2 Subjects and Ranking Procedure

Thirty-six subjects (23 male and 13 female), age ranging from 22 to 59 years (M = 35.4 years), participated in the study. They were in part staff members from Heinrich Hertz Institute, in part students, acquaintances of

[1] A file of this paper with colored pictures can be obtained at http://www.hhi.de/IM/publications

staff members or other people interested in participating in psychological experiments. 20 subjects were wearing glasses, all had normal color perception. The subjects were not familiar with the pattern set. The subjects had to indicate the perceived similarity between every combination of the 20 patterns, including the reversed combinations (same pair of patterns, but reversed target position), leading to a total of 380 coupled comparisons. In addition, the subjects had to rate randomly chosen 5% of the pattern combinations twice, adding an additional 20 comparisons. The patterns were presented on a large computer display (24'') in 40 rounds with one target stimuli and 10 other stimuli at a time. These patterns could be moved horizontally with the mouse. The subjects had to draw the targets to that distances which indicated their perceived similarity to the target stimuli (see Fig. 2): very similar stimuli had to be moved to the left near the target stimulus, very dissimilar stimuli to the right. The "similarity index" of the momentary position of a moved pattern was indicated in numbers from 1 (totally dissimilar) to 100 (totally similar).

Overall, this procedure yielded 14400 similarity ratings for the 20 patterns (36 subjects X 400 similarity judgements). Prior to statistical analysis, the ratings of the doubly rated pattern pairs were averaged, reducing the total number to 13680 judgements. 877 judgements whose deviation to the reversed similarity rating exceeded a value of 40 were considered as "wild scores" and excluded from the multidimensional scaling analysis.

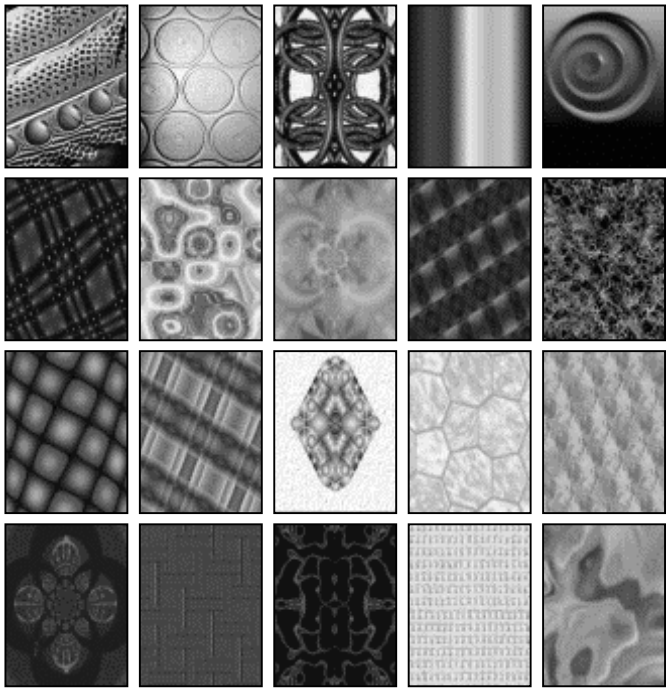


Fig. 1. Pattern set used in the experiment. Numbers (1-5, 6-10,11-15,16-20) will be referred to throughout the paper.

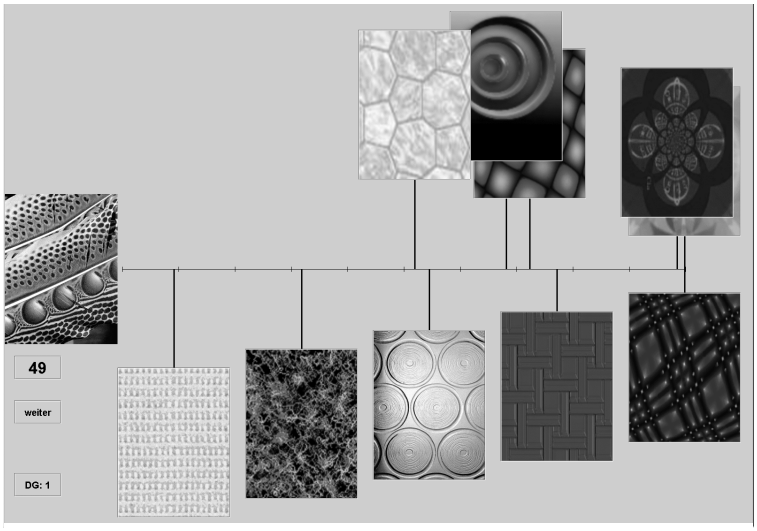


Fig 2. Screenshot of the stimuli presentation. Left: the target stimulus. The five patterns below the middle line show the initial random presentation of patterns. The five patterns above have already been moved by mouse to indicate the perceived similarity. The position of the momentarily active pattern 14 (the gray one) has a similarity value of "49", indicated in a small window on the left.

3 DATA ANALYSIS AND RESULTS

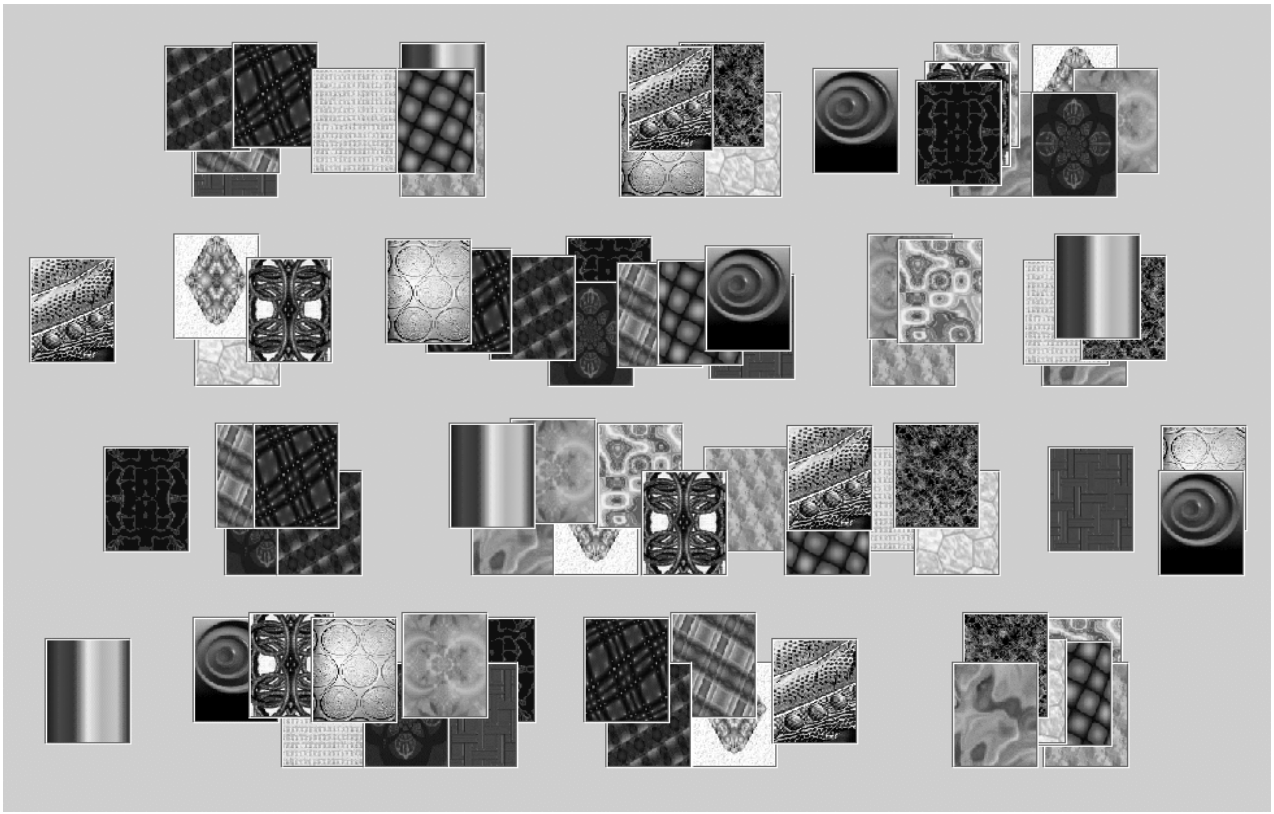


Fig. 3. The patterns, arranged according to their values in each of the four extracted dimensions. The horizontal distance between two patterns represents the dissimilarity in that dimension. Therefore, distinctive patterns for a dimension are placed on the left and right, giving the dimensional "poles" of that dimension. Uncharacteristic ("undecided") patterns are placed in the middle. See text for further discussion.

3.1 Multidimensional Scaling Results

Multidimensional Scaling is a frequently used statistical procedure in social sciences, which allows to extract underlying independent dimensions of dissimilarity from a pool of dissimilarity ratings of stimulus pairs[2]. Mojsilović et al. were able to extract up to 5 meaningfully interpretable dimensions from their data material, while the (statistically possible) six-dimensional solution turned out to be unintelligible. We used the identical statistical procedure[3] and the same statistic package (SPSS). With our data, we were able to extract up to four meaningful dimensions, while the five-dimensional and six-dimensional solutions turned out to be not interpretable in a coherent way. In the following, we will confine the discussion on the most informative four-dimensional solution, omitting their development out of the two- and three-dimensional solution. In Fig. 3, the order of the 20 patterns in each of the four dimensions is pictured.

Dimension 1: Directionality. On the left side of Dimension 1, all patterns with continuous lines are gathered. On the right side, patterns

[2] A short introduction of the underlying mathematical principles is e.g. presented in Mojsilović et al.

[3] The specifications were: weighted MDS (accounting for individual differences), squared and asymmetric matrix (using the differences between reversed similarity ratings), matrix conditionality and ordinal scale analysis level. Even if the data is in principle interval scaled, the relatively high deviations between doubly as well as reversed similarity judgements made it more plausible to treat the data as being ordinal scaled.

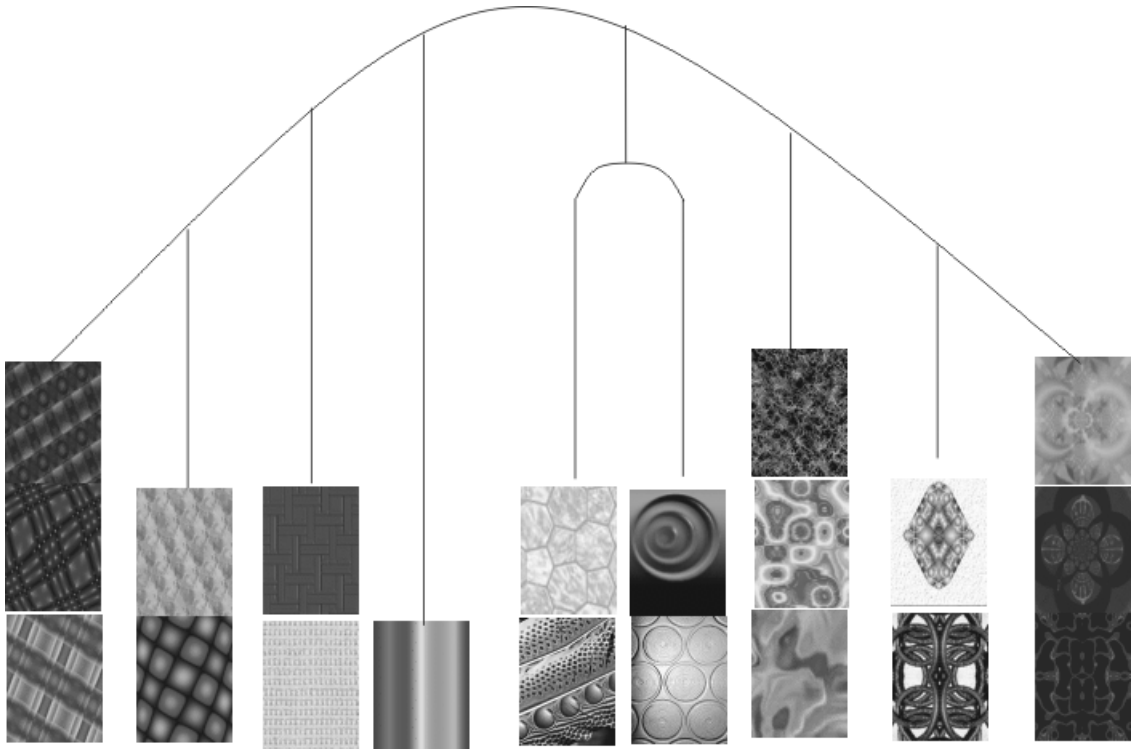


Fig. 4. Dendrogramm (simplified) of the hierarchical clustering analysis. Most similar patterns are piled up; lines indicate the combination of patterns in higher cluster.

with closed and rounded lines are placed. Three of the four "undecided" patterns in the middle are irregular and therefore neither continually lined nor rounded (1, 10, 13). In contrast, pattern 2 consists of small *round* circles which are placed in a *linear* way: this pattern belongs to both sides of this dimension. The important psychological similarity feature in this dimension can be interpreted as the directionality and orientation of the patterns.

Dimension 2: Color purity. On the left side of Dimension 2, the four black-and-white pictures can be found, followed by patterns whose colors contain a relatively high share of white or black. On the right side, the colors are becoming more and more saturated and vivid. The distinctive psychological feature behind this dimension is the color purity of the patterns.

Dimension 3: Purpleness. The seceding feature in this dimension seems to be color tone. All patterns containing red are gathered on the left side, most distinctly the patterns containing some purple (6, 9, 12, 16, 18). On the right side, all patterns without red are placed, but they don't seem to be sorted in coherent colors which could be interpreted as being most dissimilar to red (e. g. yellow<blue< green). Additional Analyses with MDS on single-case data indicated that the usage of color tone as a similarity feature is relatively common, but the impressions of dissimilarity between different color tones differ from subject to subject. Therefore, no clear cut "ranking" emerged in the global analysis. Besides, the "purpleness" as "the" distinctive feature of color tone similarity is possibly only a "majority decision"; at least some people favored green (pattern 5, 10, 15, 17) as being distinctive against all other colors.

Dimension 4. Regularity, Orientation and Complexity. Basically, the structural blueprint of the patterns in Dimension 4 changes from being symmetrical and orthogonal on the left over being diagonally oriented and of disturbed symmetry to complete asymmetry on the right. The left group consists of all patterns

from the stimulus set which are symmetric and orthogonally oriented. The middle group contains diagonally striped patterns (9, 12, 6, and 1). Besides being not orthogonally oriented, the placement of stripes in Pattern 6 and 1 is of disturbed symmetry. The right group of patterns contains all irregular shaped patterns (7, 10, 14, 20) with no obvious preferred direction.

There are three exceptions to these overall placement rules: pattern 13 is placed in the middle group though being symmetrical and orthogonal; and patterns 11 and 15 which are situated in the right group though being striped diagonally and at least somewhat symmetric. A plausible explanation for the "right shift" of these three items is their high complexity: pattern 13 is a very baroque pattern with a lot of fine details, pattern 11 possesses a complex substructure and pattern 15 is drawn in vaporous and quivering colors. In addition, the patterns to the very left are not only symmetric and orthogonal but of especially clear-cut character. Therefore, it can be concluded that the degree of complexity is the third feature underlying this dimension.

Summary. Taken together, the four dimensions account for 54% of the variance (squared correlation, *RSQ*) in the dissimilarity judgements of the subjects. Taken into account that the perception of pattern similarity is a highly idiosyncratic process which might differ from person to person, the reduction on 4 basic feature dimensions which explain more than half of the variance is a satisfying outcome.

3.2 Hierarchical Clustering While MDS allows to determine basic dimensions underlying the mass of single similarity ratings, hierarchical clustering is a statistical technique to clarify the neighborhood conditions between the rated items, in this case the patterns. Starting with all patterns singularized, the patterns with the highest similarity ratings are successively combined to clusters; adjacent patterns are combined with such clusters until finally the clusters are combined to one single cluster. The "pathway" of this clustering process (a dendrogram) pictures the relative "kinship" of the 20 patterns in the study (see Fig 4). This "kinship" can be attributed (ideally) to the combined distances of patterns in the 4 dimensions of the MDS. E.g., pattern 6, 9 and 12 are closely together on all four dimensions of the MDS; in concordance, they are combined very early in the hierarchical cluster analysis (see very left pile in fig. 4). Pattern 11 and 15 are somewhat similar to pattern 6, 9, 12, namely: they have the same directionality (continuing and diagonal lines) and a similar midlevel color purity. But both patterns are a) not purplish and b) complex res. vaporous and not clear-cut. Accordingly, pattern 11 and 15 are grouped together to an own cluster (second-left pile) which is then on a higher level combined with the cluster containing 6, 9, and 12. The other clustering processes can be acknowledged in similar ways.

4 DISCUSSION

Our study was a moderately modified attempt to replicate the findings reported by Moijsilović et al. Several important dimensions of pattern similarity judgement could successfully be replicated; namely the following: directionality, regularity, orientation, complexity and color purity. The general importance of these dimensions for pattern comparisons has been confirmed by our results. However, it should be mentioned that orientation and complexity were affiliated with regularity and not with directionality res. heaviness, as was the case in the foregoing study.

Two other dimensions from the Moijsilović et al. study could not be identified in our statistical analyses: the "pattern heaviness" and the "overall color (chromaticity)" dimensions, the latter having emerged as most important dimension in the preceding study. In addition, our MDS turned out a new dimension, the "purpleness" dimension. This is particularly interesting, since Moijsilović et al. hypothesized that color tone will generally not be a dimension for pattern comparisons. This turned out to be false: we can conclude that this dimension, as well as pattern heaviness and chromaticity, are at least sometimes relevant, but not always (probably dependent on salient features to distinguish between patterns).

Finally, another result which could not be replicated were the rules of feature combinations which reportedly accounted for the hierarchical clustering analysis in the foregoing study. Moijsilović et al. assumed that they had identified some general rule ("a grammar") of feature combinations in pattern similarity judgement. According to our findings, this is not the case. It seems more plausible that the sum of dissimilarity between two patterns on the identifiable dimensions accounts for their overall distance in the histogram. But this sum can result from any combination of dimensional differences and is not bound to a certain sequence in combining this dissimilarity values.

Conclusions. In combining the results of the two color pattern similarity studies, a more differentiated evaluation of the generally and seemingly only infrequently important pattern features has evolved, installing something like a (still incomplete) ground truth for color patterns. The next step will be to test the output of different texture descriptors against these psychological dimensions of patterns.

REFERENCES

[1] A. R. Rao and G. L. Lohse, "Toward a texture naming system: Identifiying relevant dimensions of texture", *Vis. Res.*, vol 36, no 11, pp 1649-1669, 1996.

[2] Mojsilović, , J. Kovačević, , D. Kall, , R. Safranek, , S. Ganapathy, "The vocabulary and grammar of color patterns", *IEEE Transactions on image processing*, Vol. 9, No. 3, 2000.

PROTOTYPE BASED INFORMATION RETRIEVAL IN MULTILANGUAGE BIBLES

Jarmo Toivonen[1], *Ari Visa*[1], *Tomi Vesanen*[1], *Barbro Back*[2], *and Hannu Vanharanta*[3]
[1]Tampere University of Technology, P.O. Box 553, FIN-33101 Tampere, FINLAND
e-mail: {Jarmo.Toivonen, Ari.Visa, Tomi.Vesanen}@tut.fi
[2]Åbo Akademi University, Lemminkäisenkatu 14 A, FIN-20520 Turku, FINLAND
e-mail: Barbro.Back@abo.fi
[3]Pori School of Technology and Economics, P.O. Box 300, FIN-28101 Pori, FINLAND
e-mail: Hannu.Vanharanta@pori.tut.fi

ABSTRACT

In this paper a new IR methodology based on prototype matching is presented. A prototype is an interesting document or a part of an extracted, interesting text. The prototype is matched with the existing document database or the monitored document flow. Our claim is that the new methodology is capable of automatic content-based filtering using the information of the document. To verify this hypothesis a test was designed with the Bible. Four different translations, English, Finnish, German, and Latin were selected as test text material. Verification experiments that included the search of ten nearest books to every book of the Bible were performed with a designed prototype version of the software application. The results are reported in this paper.

1 INTRODUCTION

The Internet, flatbed scanners and computers have made it possible to produce huge amounts of text documents. Now, it is essential to manage them. It is extremely important to find the desired documents. The information retrieval in text documents has usually been based on automata theories, grammars, language theories, fuzzy logic, natural language processing, or latent semantic analysis. [1, 3].
A common approach to topic detection and tracking is usage of keywords. This approach is based on assumption that the keywords given by the authors characterise the text well. This might be true but then one neglects the accuracy. More accurate method is to use all the words of a document and the frequency distribution of words. Now the comparison of frequency distributions is a complicated task. There are theories that the rare words in the histograms distinguish documents [2]. Our approach utilises this idea but in a peculiar way. The idea is expanded also to sentence and paragraph levels.

In this paper we represent our methodology and test it for the problem of information retrieval. The evolution of the methodology has been earlier discussed in several publications [6, 4, 5]. In the second chapter the applied methodology is described. In the third chapter the designed experiments are described and the validation results are reported. Finally, the methodology and the results are discussed.

2 METHODOLOGY

The methodology is based on word, sentence, and paragraph level processing. The original text is first preprocessed, extra spaces and carriage returns are omitted, etc. The filtered text is next translated into a suitable form for encoding purposes. Encoding of words is a wide subject and there are several approaches for doing it:
1) The word is recognised and replaced with a code. This approach is sensitive to new words.
2) The succeeding words are replaced with a code. This method is language sensitive.
3) Each word is analysed character by character and based on the characters a key entry to a code table is calculated. This approach is sensitive to capital letters and conjugation if the code table is not arranged in a special way.

We chose the last alternative, because it is accurate and suitable for statistical analysis. A word w is transformed into a number in the following manner:

$$y = \sum_{i=0}^{L-1} k^i * c_{L-i} \tag{1}$$

where L is the length of the character string (the word), c_i is the ASCII value of a character within a word w, and k is a constant.

Example: word is "**c a t**".

$$y = k^2 * ascii(c) + k * ascii(a) + ascii(t) \tag{2}$$

The encoding algorithm makes a different number for each different word, only the same word can have an equal number. After each word has been converted to a code number we set minimum and maximum values to words, and look the distribution of words' code numbers. Now one tries to estimate the distribution of the code numbers. Weibull distribution is selected to represent the distribution of the code numbers. Other distributions, e.g. Gamma distribution, are also possible.

However, the selected distribution should have only a few parameters and it should match the observed distribution as well as possible.

In the training phase the range between the minimum and the maximum values of words' code numbers is divided to N_w logarithmically equal bins. The count of words belonging to each bin is calculated. The bins' counts are divided with the number of all words. Then the best Weibull distribution corresponding to the data must be determined. Weibull distribution is compared with distribution by examining both distributions' cumulative distribution. Weibull's Cumulative Distribution Function is calculated by:

$$CDF = 1 - e^{(((-2.6*log(y/y_{max}))^b)*a)} \quad (3)$$

There are two parameters that can be changed in Weibull's CDF formula: a and b. A set of Weibull distributions are calculated with all the possible combinations of a's and b's using a selected precision. The possible values for the coefficients are restricted between suitable minimum and maximum values. The cumulative code number distribution and Weibull's cumulative distribution are compared in the smallest square sum sense.

In the testing phase the best Weibull distribution is found and it is now divided to N_w equal size bins. The size of every bin is $1/N_w$. Every word belongs now to a bin that can be found using the code number and the best fitting Weibull distribution. Using this type of quantisation the word can now be presented as the number of the bin that it belongs to. Due to the selected coding method the resolution will be the best where the words are most typical to text (usually 2-5 length words). Rare words (usually long words) are not so accurately separated from each other. Similarly on the sentence level every sentence has to be converted to a number. First every word in a sentence is changed to a bin number in the same way we did with words earlier. Example:

I	**have**	**a**	**cat**	**.**
bn_0	bn_1	bn_2	bn_3	bn_4

where bn_i = bin number of the word i.

The whole encoded sentence is now considered as a sampled signal. The signal is next Fourier transformed. Since the sentences of the text contain different numbers of words, the sentence vectors' lengths differ. Here we use the Discrete Fourier Transform (DFT) to transform the sentence vectors. We do not consider all the coefficients. The input for the DFT is $(bn_0, bn_1, ..., bn_n)$. DFT's outputs are coefficients B_0 to B_n. The second coefficient B_1 is selected to be the number that describes the sentence. The reason why the B_1 component is selected is that in the experiments it has been observed that B_0 is too much effected by the sentences' length.

After every sentence has been converted to numbers, a cumulative distribution is created from the sentence data set in the same way as on the word level. Now the range between the minimum and the maximum value of the sentence code numbers are divided to N_s equal size bins. The count of sentences belonging to each bin is calculated and the bins' counts are divided with the number of all sentences. The best Weibull distribution corresponding to the sentence data is found using the cumulative distribution of both distributions. Now the best distribution can be used in the quantisation of sentences. An example of a sentence distribution and a corresponding best Weibull distribution are illustrated in Fig. 1, subplots 1 and 3. In these examples the number of the bins N_s is 25.

On the paragraph level the methods are similar. The paragraphs of the document are first converted to vectors using the code numbers of the sentences. The vectors are Fourier transformed and the coefficient B_1 is chosen to represent the paragraph. After the best Weibull distribution corresponding to the paragraph data is found it can be used in the quantisation of paragraphs.

When examining the single text documents, we create histograms of the documents' word, sentence, and paragraph code numbers according to the corresponding value of quantisation. On the word level the filtered text from a single document is encoded word by word. Each word code number is quantised using word quantisation created with all the words of the data base. The right quantisation value is determined, an accumulator corresponding to the value is increased, and thus a word histogram A_w is created. The histogram A_w consisting of N_w bins is finally normalised by the word count of the document. On the sentence and the paragraph levels the histogram creation process is similar. The single document is encoded to sentence and paragraph code numbers and the hits according to the corresponding place in the quantisation are collected in histograms A_s and A_p. An example of a sentence histogram is illustrated in Fig. 1, subplot 2. With the histograms from all the documents in the database we can compare and analyse the single documents' text on the word, sentence, and paragraph levels. The histogram creation and comparison processes are illustrated in Fig. 2. Note, that it is not necessary to know anything from the actual text document to do this. It is sufficient to give one document as a prototype. The methodology gives the user all the similar documents, gives a number to the difference, or clusters similar documents.

3 EXPERIMENTS

To check how the methodology works with multilanguage documents two tests were designed. The tests were planned so that the results would depend on the information in the documents, on the language, the style, and naturally on the methodology. It was important to find a text that is carefully translated into another lan-

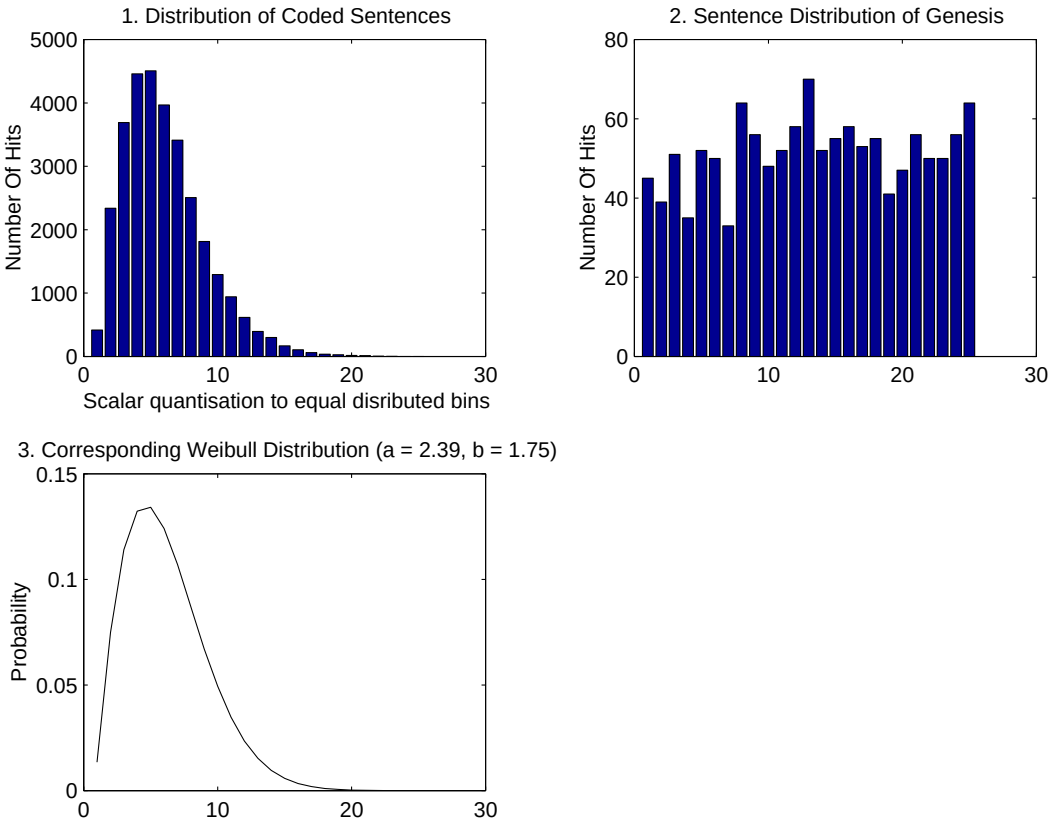


Figure 1: Example of a sentence quantisation process.

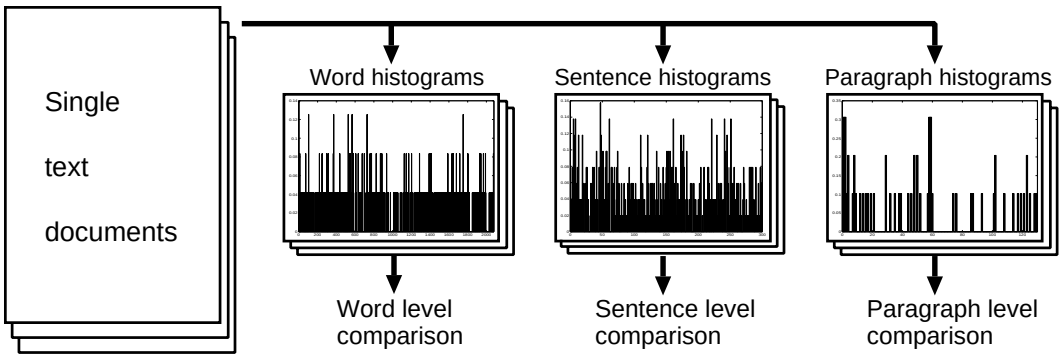


Figure 2: The process of comparing and analysing documents based on the extracted histograms on different levels.

guage. In translation it is important, at least, to keep the information even though the form depends on the language. The Bible was selected to meet the demands.

The translations used were King James Bible in English and the translations from years 1933 (the Old Testament) and 1938 (the New Testament) in Finnish. The German version is the Luther translation, and the Latin version Jerome's translation (Vulgate) from years 382-405. The idea was to select a window of closest matches and to compare all the books in the Bible. The bin size, and thus the size of the histograms, for the word level was 2080, for the sentence level 25, and for the paragraph level 10. The word, the sentence and paragraph level histograms were created based on the whole text of the Bibles in four languages. Euclidean distance was used in their comparison.

In the first experiment the capability of the methodology to separate documents on a coarse level was examined. We know that the Old and the New Testament books differ and expected to see a difference in the first test. Every book was one by one taken as a prototype document, and ten closest matches were examined. Note, that the order within the window is not considered, only the co-occurrences. The number of books in the window that matched with other books in the Old Testament, respectively in the New Testament are reported for four languages in Tables 1, 2, 3, and 4. For example, for the Genesis (book number 1) in English, we see that on the word and the sentence level there are six Old Testament books among the ten closest books, and on the paragraph level there are nine.

The idea of the second experiment was to compare ten closests matches in each language and count the co-occurrences. The results on three different levels are shown in Tables 5, 6, and 7. Let's examine again the first book in the Bible, the Genesis. When the ten closest book in each language are collected the total number of books is 40. For the Genesis, among these 40 books there are five books that appear only once, 2 books that appear twice, five books that appear three times, and four books that appear in all languages.

4 DISCUSSION

The main idea is to test the ability to find similar contents. One of our basic assumptions is that within a specific field, for instance in law or business, the ambiguities of words will not disturb. Our experiments are based on the model that the content of a document is described by the information, the language, and the style. That was the reason why the Bible was selected as test material. We know that the translations have been done very carefully, at least at the information level. The influence of language and the style is eliminated by using English, Finnish, German, and Latin versions of the Bible. First a simple test was designed: the task was to distinguish between the Old Testament and the New Testament. The search was done by taking one book as a prototype and all similar books to that book were searched. Ten closest matches were displayed and all the books were checked. The results were similar from language to language. At the word level on average eight books from ten were from the assumed class. At the sentence level there was more variety, from five to six books were from the assumed class. At the paragraph level on average five books from ten were from the assumed class It was amazing to note that independently of language it was possible to find four same books within ten closest matches at the word level. At the sentence level independently of language it was possible to find two same books within ten closest matches but at the paragraph level only 0.4 books were possible to find. These results are far better than is possible to achieve by random.

It seems that a methodology capable of content-based filtering has been developed. The methodology can easily be adapted to new fields by training.

5 ACKNOWLEDGMENTS

The financial support of TEKES (grant number 40943/99) is gratefully acknowledged.

References

[1] F. C. Gey. Information Retrieval: Theory, Application, Evaluation. In *Tutorial at the Thirty-Third Annual Hawaii International Conference on System Sciences (HICSS-33)*, January 4–7 2000.

[2] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[3] D. W. Oard and G. Marchionini. A conceptual framework for text filtering. Technical Report CS-TR3643, University of Maryland, May 1996.

[4] A. Visa, J. Toivonen, S. Autio, J. Mäkinen, H. Vanharanta, and B. Back. Data Mining of Text as a Tool in Authorship Attribution. In *Proceedings of AeroSense 2001, SPIE 15th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls.*, 2001. To be published.

[5] A. Visa, J. Toivonen, B. Back, and H. Vanharanta. Improvements on a Knowledge Discovery Methodology for Text Documents. In *Proceedings of SSGRR 2000 – International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, L'Aquila, Italy, July 31–August 6 2000. (CD–ROM).

[6] A. Visa, J. Toivonen, H. Vanharanta, and B. Back. Prototype Matching – Finding Meaning in the Books of the Bible. In *Proceedings of the Thirty-Fourth Annual Hawaii International Conference on System Sciences (HICSS-34)*, January 3–6 2001. (CD–ROM).

Table 1: Number of books corresponding to the prototype's Testament among ten closest matches in English.

The Old Testament, book number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
Word	6	10	10	10	10	10	9	5	6	8	10	10	9	10	10	10	10	8	10	9	6	9	10	10	9	10	9	10	10	10	10	6	10	10	10	10	10	10	10
Sentence	6	4	9	5	9	8	9	9	9	9	8	9	6	8	7	8	7	5	5	7	5	5	7	9	5	10	9	8	10	6	9	6	2	6	7	5	9	10	7
Paragraph	9	8	7	5	8	5	9	6	10	8	7	8	3	9	8	8	5	9	9	8	7	8	9	8	9	10	7	7	7	8	7	6	8	7	7	7	5	9	5

The New Testament, book number	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
Word	5	4	5	5	4	10	9	10	10	9	5	10	10	10	10	10	10	9	10	10	10	7	9	10	7	9	0
Sentence	6	5	5	5	3	6	5	3	3	7	6	3	4	6	3	8	5	5	5	6	1	6	5	5	3	5	0
Paragraph	4	2	3	2	3	3	1	2	1	1	4	3	1	4	3	3	2	4	1	4	2	6	4	4	4	6	2

	The Old Testament average	The New Testament average	Total average
Word	9.21	8.04	8.73
Sentence	7.23	4.59	6.15
Paragraph	7.44	2.93	5.59

Table 2: Number of books corresponding to the prototype's Testament among ten closest matches in Finnish.

The Old Testament, book number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
Word	10	10	10	10	10	10	10	8	10	10	10	10	10	10	9	10	10	10	10	9	8	8	10	10	8	10	9	10	9	10	10	10	10	9	10	10	10	10	10
Sentence	6	7	10	9	8	7	9	9	7	8	7	10	7	10	9	6	7	6	5	7	4	7	6	9	4	9	9	6	5	9	6	3	5	4	5	7	6	9	7
Paragraph	4	9	7	4	8	5	8	9	7	6	8	7	4	9	5	6	6	9	9	8	7	7	8	8	6	9	8	5	6	8	6	9	8	5	7	7	7	8	5

The New Testament, book number	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
Word	5	5	4	8	4	10	10	10	10	10	5	10	10	10	10	10	10	6	9	10	10	9	10	10	10	7	3
Sentence	4	4	3	4	1	3	4	2	3	2	4	5	4	3	4	5	1	3	1	5	4	2	4	3	2	3	0
Paragraph	5	3	4	4	2	4	4	4	2	3	5	4	4	3	3	3	4	4	1	2	5	4	2	3	3	4	2

	The Old Testament average	The New Testament average	Total average
Word	9.67	8.33	9.12
Sentence	7.03	3.07	5.41
Paragraph	6.97	3.37	5.50

Table 3: Number of books corresponding to the prototype's Testament among ten closest matches in German.

The Old Testament, book number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
Word	7	10	10	9	10	9	10	7	10	10	10	9	9	9	9	9	10	8	10	6	9	8	10	10	10	9	9	8	9	10	10	10	10	10	10	10	10	10	9
Sentence	7	5	9	6	9	9	10	8	9	9	8	9	7	9	9	8	8	5	5	4	5	6	6	10	4	9	10	8	7	8	8	2	7	5	4	6	9	6	7
Paragraph	8	7	7	5	7	7	9	7	9	5	8	7	7	8	5	7	6	7	9	8	8	4	9	9	8	9	6	4	6	7	7	7	9	5	6	6	7	6	8

The New Testament, book number	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
Word	5	4	4	9	4	9	9	10	9	6	5	10	10	10	8	8	9	9	5	7	9	8	10	10	6	9	0
Sentence	5	4	4	4	4	4	4	5	4	7	5	4	5	2	6	5	5	4	2	7	6	2	3	6	6	1	0
Paragraph	3	2	3	4	2	4	1	4	4	3	3	4	4	2	3	3	5	6	4	4	1	3	2	5	4	4	1

	The Old Testament average	The New Testament average	Total average
Word	9.28	7.48	8.55
Sentence	7.18	4.22	5.97
Paragraph	7.03	3.26	5.48

Table 4: Number of books corresponding to the prototype's Testament among ten closest matches in Latin.

The Old Testament, book number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
Word	8	10	9	8	10	10	9	7	8	10	10	10	10	10	10	9	9	9	10	5	9	10	10	10	7	9	9	9	10	10	10	8	10	10	10	10	10	9	9
Sentence	5	6	6	5	6	9	9	10	9	8	8	10	5	9	9	8	9	4	4	4	5	5	6	9	6	9	9	5	8	7	3	5	8	7	5	9	5	9	8
Paragraph	8	8	6	7	9	9	5	5	6	6	2	9	5	8	9	7	8	7	10	5	8	7	9	8	7	7	6	7	5	7	6	6	7	5	5	7	7	7	7

The New Testament, book number	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
Word	4	4	3	4	3	10	8	10	9	9	4	8	9	10	8	6	6	6	6	6	10	9	7	5	8	8	2
Sentence	3	5	5	4	2	5	5	4	4	2	3	2	4	5	4	5	3	3	2	6	3	2	0	3	3	1	1
Paragraph	6	5	6	3	5	1	3	2	0	2	6	3	2	3	2	5	4	2	1	2	3	2	4	5	5	5	1

	The Old Testament average	The New Testament average	Total average
Word	9.23	6.74	8.21
Sentence	6.95	3.30	5.45
Paragraph	6.85	3.26	5.38

Table 5: Number of co-occurrences among the ten closest matches in four languages, the word level.

The Old Testament, book number

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
1	5	5	9	8	7	5	1	9	4	4	0	2	4	6	6	4	4	5	3	14	13	13	7	7	9	6	8	9	11	4	9	11	6	8	8	4	5	10	10
2	2	5	7	3	4	2	2	4	7	3	3	3	3	6	1	2	1	4	2	3	5	3	4	5	3	4	2	3	2	3	4	5	3	7	7	5	7	4	2
3	5	3	3	2	3	5	5	1	2	2	2	4	2	2	4	4	2	5	3	4	3	3	3	1	3	2	4	3	3	2	1	5	4	2	2	2	3	2	2
4	4	4	2	5	4	4	5	5	4	6	7	5	6	4	5	5	7	3	6	2	2	3	4	5	4	5	4	4	4	6	5	1	4	3	3	5	3	4	5

The New Testament, book number

	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
1	8	3	4	10	7	2	5	2	2	13	6	6	4	3	7	10	11	10	11	11	2	8	7	12	12	10	6
2	3	6	5	4	5	2	3	4	3	2	1	4	1	6	3	6	3	6	6	3	3	5	3	4	5	4	7
3	2	3	2	2	1	2	3	2	0	1	4	2	6	3	1	2	5	2	3	5	4	2	5	4	6	2	4
4	5	4	5	4	5	7	5	6	8	5	5	5	4	4	6	3	2	3	2	2	5	4	3	2	0	4	2

	The Old Testament average	The New Testament average	Total average
1	6.74	7.11	6.89
2	3.72	3.96	3.82
3	2.90	2.89	2.89
4	4.28	4.07	4.20

Table 6: Number of co-occurrences among the ten closest matches in four languages, the sentence level.

The Old Testament, book number

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
1	6	5	4	3	4	13	5	9	6	6	6	6	9	5	11	11	9	5	5	10	7	14	5	5	8	5	1	10	23	14	20	19	13	15	12	12	16	6	13
2	2	3	8	5	8	5	4	9	7	4	5	1	5	2	6	9	9	4	4	3	7	6	6	2	4	3	3	6	7	5	7	6	9	9	5	11	6	5	4
3	6	3	4	5	4	3	5	3	4	6	4	4	3	5	3	1	3	1	1	4	1	2	5	5	4	3	7	2	1	4	2	3	3	1	6	2	4	4	5
4	3	5	2	3	2	2	3	1	2	2	3	5	3	4	2	2	1	6	6	3	4	2	2	4	3	5	3	3	0	1	0	0	0	1	0	0	0	3	1

The New Testament, book number

	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
1	4	10	6	5	8	6	6	8	10	17	10	16	9	21	20	7	19	24	7	14	15	19	15	26	28	16	6
2	4	4	4	3	1	5	6	7	4	5	8	6	8	8	7	5	9	5	7	5	5	9	8	7	6	9	4
3	4	2	2	3	6	4	2	6	6	3	2	4	5	1	2	5	1	2	5	4	5	1	3	0	0	2	2
4	4	4	5	5	3	3	4	0	1	1	2	0	0	0	0	2	0	0	1	1	0	0	0	0	0	0	5

	The Old Testament average	The New Testament average	Total average
1	9.13	13.04	10.73
2	5.49	5.89	5.65
3	3.49	3.04	3.30
4	2.36	1.52	2.02

Table 7: Number of co-occurrences among the ten closest matches in four languages, the paragraph level.

The Old Testament, book number

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
1	13	13	13	6	18	8	10	27	13	10	13	17	6	16	13	15	21	10	7	15	13	25	11	11	21	8	16	22	15	16	18	18	19	23	25	16	25	15	21
2	9	9	7	6	6	8	7	5	8	6	9	7	7	9	9	6	6	7	7	6	9	6	6	8	8	9	9	6	11	6	8	9	6	7	6	12	6	7	5
3	3	3	3	6	2	4	4	1	1	6	3	3	4	2	3	3	1	4	5	3	3	1	3	3	1	2	2	2	1	4	2	0	3	1	1	0	1	1	3
4	0	0	1	1	1	1	1	0	2	0	0	0	2	0	0	1	1	1	1	1	0	0	2	1	0	2	0	0	0	0	0	1	0	0	0	0	0	2	0

The New Testament, book number

	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
1	13	13	13	15	10	17	17	20	24	21	24	19	19	19	21	28	27	28	21	24	25	26	22	21	23	29	8
2	5	10	9	8	9	7	10	4	8	8	8	9	9	6	6	6	5	6	5	8	6	7	6	5	7	4	10
3	3	1	3	3	4	3	1	4	0	1	0	1	1	3	1	0	1	0	3	0	1	0	2	3	1	1	4
4	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

	The Old Testament average	The New Testament average	Total average
1	15.44	20.26	17.41
2	7.38	7.07	7.26
3	2.51	1.67	2.17
4	0.56	0.15	0.39

AN EFFICIENT SCHEME FOR AUTOMATIC VOP-BASED ORGANIZATION OF STEREO-CAPTURED VIDEO SEQUENCES

Klimis S. Ntalianis, Nikolaos D. Doulamis, Anastasios D. Doulamis, Georgios Patikis, and Stefanos D. Kollias
Dept. of Electrical & Computer Engineering, National Technical University of Athens,
Heroon Polytechniou 9, 157-73, Zografou, Athens, GREECE
Tel: +30-1-772 2488; Fax: +30-1-772 2492
E-mail: kntal@image.ntua.gr

ABSTRACT

An efficient scheme for automatic VOP-based organization of stereoscopic sequences is proposed in this paper, which produces a graph-like structure of the sequence. The scheme is oriented in generating correlation links between "key"-VOPs of different shots, so that the provided structure can be used for effective video indexing or fast VOP browsing. In particular, initially scene change detection is performed and for each frame of a shot, a feature vector is constructed. In the next phase key-frames within each shot are extracted, using an optimization method for locating frames of minimally correlated feature vectors. The optimization problem is tackled by a genetic algorithm approach. Afterwards for each key-frame an unsupervised color/depth segmentation fusion algorithm is incorporated to extract the "key"-VOPs within this frame. Finally correlation links are generated between "key"-VOPs to produce a graph-like structure of the video sequence. Experimental results on real life stereoscopic video sequences indicate the promising performance of the proposed scheme.

1. INTRODUCTION

Recent progress in the fields of video capturing, encoding and processing has led to an explosion in the amount of visual information being stored and accessed. Moreover the rapid development of video-based multimedia and Internet applications has stimulated the need for efficient tools towards representation, searching and content-based retrieving.

Traditionally, video sequences are represented by numerous consecutive frames (stereo pairs in the case of stereoscopic video) each of which corresponds to a constant time interval. However, this image sequence representation, which stems from the analog tape storage process, results in a linear (sequential) access to the video content

While this approach may be adequate for viewing a video in a movie mode, it raises a number of limitations when addressing new applications, such as fast video browsing, content-based indexing and retrieving or summarization. Currently, the only way to browse a video is to sequentially scan video frames, which is both a time-consuming and tedious process. Towards this direction, temporal reduction of the video content was proposed by employing the so-called summarization schemes, some of which are presented in [1], [2], and [3]. However these techniques do not take into consideration the semantically meaningful content of a scene (VOPs) and use the global color, motion and texture characteristics of a frame. Furthermore by these techniques a significant number of frames is discarded, some of which may contain the specific visual information a user looks for.

On the other hand results provided by current visual information retrieval systems are not yet completely satisfactory. Some interesting works include [4] and [5], where in the first work VOPs are extracted according to motion homogeneity. In the other work only global visual characteristics are considered. Another scheme for video representation is presented in [6] where low-level video content description is automatically built. However, the aforementioned schemes face two problems: (i) Except of the case in [4], the other schemes do not consider VOPs, which are usually the visual components a user searches for in an indexing application. (ii) Even after building a visual features library, every new visual search can be very much time-consuming since all feature vectors should be checked.

In this paper the proposed system innovatively organizes stereoscopic video sequences in an automatic operational mode, based on VOP information. The system produces a graph-like structure of the sequence, which can be used for selective transmission, fast browsing or directive indexing. The scheme is based on a stereo-capturing system, since depth can be reliably estimated from a stereoscopic pair of frames, and considering that each VOP occupies a specific depth [7]. After shot cut detection, for each frame of a shot a feature vector is constructed. Then within each shot, those frames having minimally correlated feature vectors are selected as key-frames. However due to the large size of the solutions space, a genetic approach is adopted as in [8]. Afterwards for each key-frame a segmentation fusion algorithm is incorporated to extract the "key"-VOPs contained in this frame. Towards this direction color and depth segments provided by a segmentation algorithm, are properly fused. Finally correlation links among "key"-VOPs are generated to provide the graph-like structure of the sequence. An overview of the proposed system is presented in Figure 1.

2. KEY FRAMES EXTRACTION

The first step towards VOP-based video organization includes shot cut detection. In our approach the algorithm

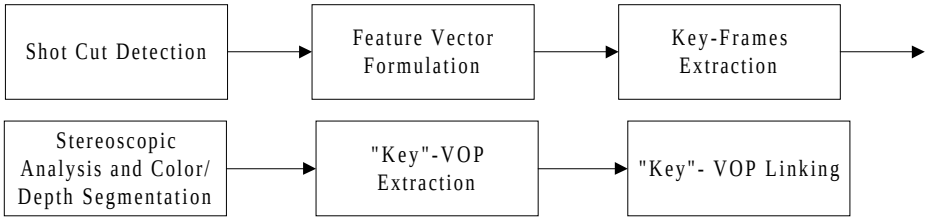


Figure 1: Overview of the proposed VOP-based video sequence organization scheme.

proposed in [9] is adopted due to its efficiency and low computational complexity. After shot cut detection for each frame of a shot, a multidimensional feature vector is constructed as in [8]. Let us denote by $\mathbf{f}_i^{L_m} \in \Re^M$, $i \in V = \{1,\ldots,N_F\}$ the feature vector of the i-th frame of the shot L_m, where $N_F{}^{L_m}$ is the total number of frames of the shot.

Let us now assume that the K_F most characteristic frames should be selected from a given shot. In this paper the key-frames are selected as the ones with the minimum correlation among a scene. This "minimum correlation" approach is adopted as we expect that the most uncorrelated frames would contain all the available different visual information among a shot.

Towards this direction we first define the *index* vector $\mathbf{a} = (a_1,\ldots,a_{K_F}) \in U \subset V^{K_F}$ where

$$U = \{(a_1,\ldots,a_{K_F}) \in V^{K_F} : a_1 < \cdots < a_{K_F}\} \quad (1)$$

is the subset of set V^{K_F}, which contains all sorted index vectors $\mathbf{a}$. Each index vector $\mathbf{a}$ corresponds to a set of frame numbers. Then the *correlation measure* of K_F feature vectors $\mathbf{f}_i^{L_m}$, $i = a_1,\ldots,a_{K_F}$ is defined as

$$R_F(\mathbf{a}) = \frac{2}{K_F(K_F-1)} \sum_{i=1}^{K_F-1} \sum_{j=i+1}^{K_F} (\rho_{a_i,a_j})^2 \quad (2)$$

where ρ_{a_i,a_j} is the correlation coefficient of feature vectors $\mathbf{f}_{a_i}{}^{L_m}$ and $\mathbf{f}_{a_j}{}^{L_m}$. As observed, the goal is to find the index vector $\mathbf{a}$ that minimizes $R_F(\mathbf{a})$.

Unfortunately, an exhaustive search within a shot for detecting the minimum value of $R_F(\mathbf{a})$ is practically unfeasible, since the multidimensional space U includes all possible combinations of frames. For this reason, a *genetic algorithm* (GA) approach is used in this paper.

In this approach, chromosomes whose genetic material consists of frame numbers represent possible solutions of the optimization problem. An *initial population* of P chromosomes, $\mathbf{A}(0) = (\mathbf{a}_1,\ldots,\mathbf{a}_P)$ is first generated using a temporal variation approach [8], which exploits the temporal relation of feature vectors. Population $\mathbf{A}(0)$ is then used for the creation of new generation populations $\mathbf{A}(n)$, $n>0$. The creation of $\mathbf{A}(n)$ at generation (or GA cycle) n is performed by applying a set of operations on population $\mathbf{A}(n-1)$. In particular, the correlation measure $R_F(\mathbf{a})$ is used as an *objective function* to estimate the performance of all chromosomes in population $\mathbf{A}(n-1)$ and *parent selection* is applied so that a fitter chromosome has a higher chance of survival in the next generation. A set of new chromosomes (offspring) is then produced by mating the selected parent chromosomes and applying a *crossover operator*, which randomly combines parental genetic material to produce the genetic material of the offspring. *Mutation* is also applied, introducing random gene variations that are useful for restoring lost genetic material, or for producing new material that corresponds to new search areas. Population $\mathbf{A}(n)$ is thus formed by inserting new chromosomes into $\mathbf{A}(n-1)$ and deleting an appropriate number of older chromosomes, following a *replacement strategy* so that each population consists of P members. This procedure is repeated in an iterative way, until $\mathbf{A}(n)$ converges to an optimal solution of the problem [10],[11].

Results of the proposed key-frames extraction module are presented in Figure 2. In this figure four key-frames for one shot of the stereoscopic sequence program "Eye to Eye" have been extracted. The selected group of frames minimizes the correlation measure $R_F(\mathbf{a})$, where in this case $\mathbf{a}$=(12, 25, 42, 71) contains the indices of the extracted frames within the shot. The key-frames of each shot are the only input elements to the next unsupervised VOP segmentation module of the proposed scheme.

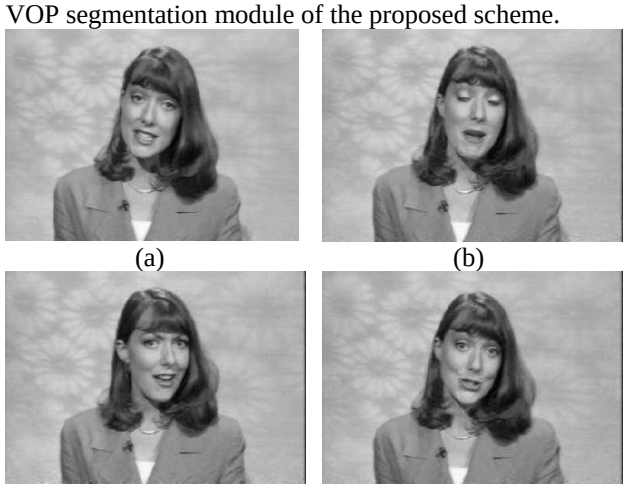


Figure 2: Four key-frames extracted from a shot of the "Eye to Eye" stereoscopic program, containing the most uncorrelated visual content within the shot. (a) The first key-frame (#8112) (b) The second key-frame (#8125) (c) The third key-frame (#8142) and (d) The fourth key-frame (#8171). Numbers inside brackets denote the frame-number among the video sequence.

3. UNSUPERVISED VOP SEGMENTATION

When key-frames for all shots of a video sequence are extracted the fourth and fifth modules of Figure 1 are activated.

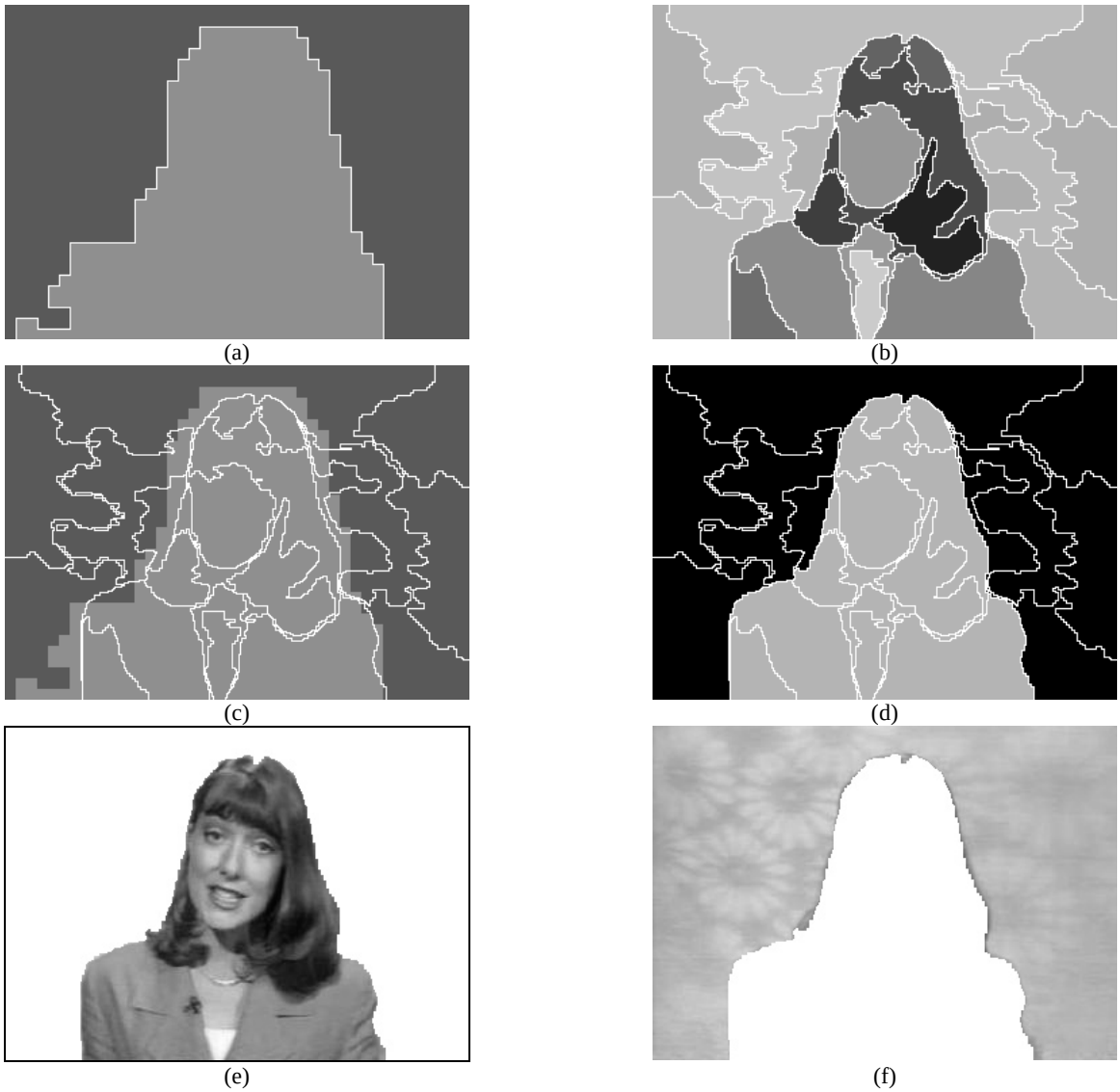


Figure 3: Segmentation fusion results for the key-frame of Figure 2(a) (first key-frame of a shot) (a) Depth segments map (b) Color segments map (c) Projection of the color segments map onto the depth segments map (d) Fusion of the color segments belonging to the same depth segment. (e) Extracted foreground video object (f) Extracted background video object.

During these modules all key-frames are analyzed and the "key"-VOPs are extracted.

Generally VOP segmentation remains a very interesting research problem, despite the extensive attention it has received the last decade. Even if it has been partly solved in semi-automatic schemes [12], there are several cases where this solution is not suitable considering high time consumption or extensive human intervention. Characteristic examples are sequences captured without using the chroma-key technology. For these reasons an unsupervised VOP segmentation technique is employed, based on fusion of color segments according to depth similarity criteria. The color/depth segments fusion scheme is based on the idea that color segments describe accurately the boundaries of a VOP. However a VOP usually consists of multiple color segments. On the other hand it has been stated that all parts of a video object are usually located on the same depth [7]. Under this consideration each depth segment roughly describes a VOP and a constrained fusion of color segments according to depth similarity provides VOPs with accurate contours.

More particularly let us consider that for a given key-frame an occlusion compensated depth map is generated, by applying the stereo pair analysis technique presented in [13]. Let as also consider that a segmentation algorithm is applied onto the aforementioned depth map and onto one of the stereo channels (left/right), producing the depth segments map and the color segments map respectively. In the current scheme a multiresolution implementation of the Recursive Shortest Spanning Tree (RSST [14]) algorithm, called M-RSST [15] is used both for color and depth segmentation. In this implementation, the RSST algorithm is recursively applied to images of increasing resolution. This approach, apart from accelerating the segmentation procedure, also reduces the number of small objects, which is a useful property in the context of VOP-based video organization.

Let us now assume that the color segments map consists of K^c color segments while the depth segments

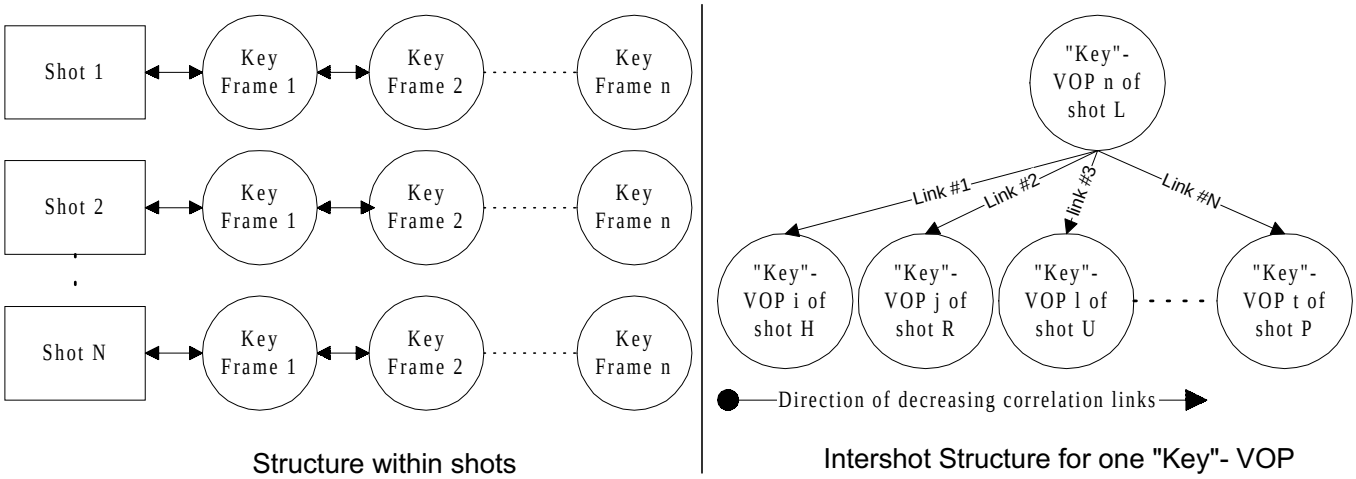


Figure 4: The proposed video sequence structure. On the left part the intrashot structure can be observed, the main elements of which are the extracted key-frames. On the right part the intershot structure for one "key"-VOP is depicted. Starting from one "key"-VOP other nodes of the tree can be visited belonging to different shots.

map includes K^d depth segments denoted as S_i^c, $i=1,2,\ldots,K^c$ and S_i^d, $i=1,2,\ldots,K^d$ respectively. The segments S_i^c and S_i^d are mutually exclusive, i.e., $S_i^c \cap S_k^c = \varnothing$ for any $i,k=1,2,\ldots,K^c$, $i \neq k$ and, similarly, $S_i^d \cap S_k^d = \varnothing$ for any $i,k=1,2,\ldots,K^d$, $i \neq k$. Let us also denote by G^c and G^d the output masks of color and depth segmentation, which are defined as the sets of all color and depth segments respectively:

$$G^c = \{S_i^c, i=1,2,\ldots,K^c\}, \quad G^d = \{S_i^d, i=1,2,\ldots,K^d\} \tag{3}$$

Color segments are projected onto depth segments so that video objects provided by depth segmentation are retained and, at the same time, object boundaries given by color segmentation are accurately extracted. For this reason, each color segment S_i^c is associated with a depth segment, so that the area of intersection between the two segments is maximized. This is accomplished by means of a *projection function*:

$$p(S_i^c, G^d) = \arg\max_{g \in G^d}\{a(g \cap S_i^c)\}, \quad i=1,2,\ldots,K^c \tag{4}$$

where $a(\cdot)$ is the area, i.e., the number of pixels, of a segment. Based on the previous equation, K^d sets of color segments, say C_i, $i=1,2,\ldots,K^d$, are defined, each of which contains all color segments that are projected onto the same depth segment S_i^d:

$$C_i = \{g \in G^c : p(g, G^d) = S_i^d\}, \quad i=1,2,\ldots,K^d \tag{5}$$

Then, the final segmentation mask, *G*, consists of $K = K^d$ segments S_i, $i=1,2,\ldots,K$, each of which is generated as the union of all elements of the corresponding set C_i:

$$S_i = \bigcup_{g \in C_i} g, \quad i=1,2,\ldots,K \tag{6}$$

$$G = \{S_i, i=1,2,\ldots,K\} \tag{7}$$

In other words, color segments are merged together into $K = K^d$ new segments according to depth similarity. The final segmentation consists of these segments, which contain the same image regions as the corresponding depth segments, but with accurate contours obtained from color segments.

VOP extraction results are presented in Figure 3 for the first key-frame of the analyzed shot, depicted in Figure 2(a). Depth segmentation, shown with two different gray levels as in Figure 3(a), is overlaid with the white contours of the color segments, as obtained from Figure 3(b). The result can be shown in Figure 3(c). Moreover, fusion of the color segments that belong to the same depth plane are presented in Figure 3(d) while the final VOP segmentation results are depicted in Figures 3(e) and 3(f) for the foreground and the background video objects respectively. As is observed, the segmentation fusion module provides video objects with accurately detected boundaries.

4. "KEY"-VOP LINKING AND APPLICATIONS

After extracting the "key"-VOPs for each key-frame, the scheme proceeds to the final step of video sequence structuring. A link is generated between each pair of "key"-VOPs according to a correlation measure. More particularly let us assume that K_{FS} "key"-VOPs have been extracted from the key-frames of the whole video sequence. Let us also denote by $\mathbf{K}_{V_i}^{L_m}$ the feature vector of the i-th "key"-VOP of shot L_m, formed in a similar way as vector $\mathbf{f}_i^{L_m}$ in section 2. Then correlation links are generated for all possible pairs of "key"-VOPs, except those pairs where both "key"-VOPs belong to the same shot. The correlation coefficient for two feature vectors

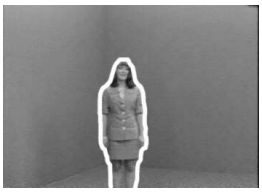
#8112

#5940

#7802

#116

#2235



#11092

Figure 5: Intershot structure for the foreground "key"-VOP as depicted in Figure 3(e). Only the first five minimum correlation key-frames are presented containing the respective "key"-VOPs. The numbers below the frames expresses the frame-number among the video sequence "Eye to Eye".

$\mathbf{K}_{V_i}{}^{L_m}$ and $\mathbf{K}_{V_j}{}^{L_q}$ with $m \neq q$, is estimated by

$$\rho_{i^m j^q} = C_{i^m j^q} \Big/ (\sigma_{i^m} \sigma_{j^q}) \quad (8)$$

where $C_{i^m j^q} = (K_{V_i}{}^{L_m} - \mathbf{m})^T (K_{V_j}{}^{L_q} - \mathbf{m})$ is the covariance of the two vectors, $\mathbf{m} = \sum_{\substack{e=1 \\ (r \in L_e)}}^{N} K_{V_r}{}^{L_e} / K_{FS}$ is the average feature vector of the K_{FS} "key"-VOPs, N is the number of shots and $\sigma^2_{i^m} = C_{i^m i^m}$ is the variance of $\mathbf{K}_{V_i}{}^{L_m}$. When correlation links are estimated for all allowed pairs, correlation values for each "key"-VOP are sorted and a graph-like structure of the sequence is generated. In Figure 4 the provided sequence structure is presented. In particular the left part depicts the sequence structure within shots, while in the right part the sorted correlation links for a "key"-VOP of a key-frame can be observed.

The provided video sequence structure can greatly benefit contemporary applications. For example in an indexing and retrieval scheme the provided visual information to be indexed can initially be compared to the "key"-VOPs of the whole sequence. A distance between "key"-VOPs and provided visual information can be incorporated, e.g. distance according to correlation, convolution, Euclidean etc. Then the following content can be retrieved: (a) The "key"-VOP with the minimum distance, (b) The shot where the minimum distance "key"-VOP originates from and (c) The first K minimum correlation "key"-VOPs provided by parsing the graph-like structure, where K is a threshold. It is evident that indexing schemes can be greatly benefited by the provided video sequence structure, as visual searching can be directively performed, which accelerates the process (compared to searching the whole feature vectors library).

Furthermore summarization schemes can benefit from the proposed automatic VOP-based sequence organization too. Firstly a summary of the whole sequence can consist of a subset of the, say, K_{VF} key-frames, extracted from the whole sequence (e.g. selection of one key-frame per shot). Additionally, in an interactive summarization scheme, a user may select a "key"-VOP out of the K_{FS} available, asking for a summary based on the selected content. Then the application can return in a hierarchical manner: (a) the other "key"-VOPs that belong to the same shot with the selected "key"-VOP, (b) the whole shot that contains the selected "key"-VOP and (c) the C minimum correlation "key"-VOPs belonging to other shots, by simply parsing the graph structure (content directive summarization).

In Figure 5 the intershot structure for the foreground "key"-VOP depicted in Figure 3(e) is presented. Only the first five minimum correlation key frames are illustrated. For presentation purposes a white line represents the boundary between the foreground video object and the background. As observed, even if there are sufficient changes in the background object, the foreground information remains the same. Furthermore as the correlation coefficient ρ for a "key"-VOP decreases, the irrelevance of the visual content retrieved increases, considering the "key"-VOP under investigation.

5. Conclusions and Discussion

The traditional video sequence representation raises a number of limitations when addressing contemporary applications, such as video browsing, content-based indexing and retrieving or summarization. These multimedia applications can be greatly benefited if efficient video sequence structuring tools and systems are developed. Under this consideration we propose an automatic system for VOP-based video sequence organization. Initially the whole video sequence is analyzed and a feature vector is constructed for each frame, turning the frame-based video representation to feature vector-based. Afterwards a shot cut detection algorithm is adopted and for each shot key-frames are extracted by minimizing a correlation measure. Due to the large dimension of possible solutions (combinations of frames) a genetic algorithm approach is incorporated, which converges very fast to an optimal solution. Then for each key-frame the semantically meaningful content is extracted using a color/depth segmentation fusion

algorithm. The procedure results into the detection of "key"-VOPs. Finally a correlation link is estimated between each pair of "key"-VOPs, excluding those pairs where both "key"-VOPs originate from the same shot. The created graph-like structure of the video sequence can be used by indexing and summarization schemes providing promising results. In future works other schemes should be investigated, taking into consideration speech and sound (generation of similar graph structures for acoustic elements). Then multimedia information retrieval or browsing can be performed based on an inter-linked graph, composed of both visual and acoustic element graphs.

6. Acknowledgements

The authors wish to thank Mr. Chas Girdwood, project manager of ITC (Winchester), for providing the 3D video sequence "Eye to Eye", produced in the framework of the ACTS MIRAGE project. Furthermore the authors want to thank Dr. Siegmund Pastoor of the HHI (Berlin), for providing the video sequences of the DISTIMA project. This research is funded by the Foundation of Governmental Scholarships of Greece.

7. References

[1] M. Mills, J. Cohen and Y. Y. Wong, "A Magnifier Tool for Video Data," *Proc. ACM Computer Human Interface (CHI)*, May 1992, pp. 93-98.

[2] S. W. Smoliar and H. J. Zhang, "Content-Based Video Indexing and Retrieval," *IEEE Multimedia*, pp.62-72, Summer 1994.

[3] M. M. Yeung and B.-L. Yeo, "Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 7, No. 5, pp. 771- 785, Oct. 1997.

[4] S. -F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A Fully Automated Content-Based Video Search Engine Supporting Spatiotemporal Queries," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 602-615 , Sept. 1998.

[5] G. Iyengar and A. B. Lipmann, "Videobook: An experiment in characterization of video, " *in Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 855-858, 1996.

[6] Y. Deng, and B. S. Manjunath, "NeTra-V: Toward an Object-Based Video Representation," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 616-627, Sept. 1998.

[7] L. Garrido, F. Marques, M. Pardas, P. Salembier and V. Vilaplana, "A Hierarchical Technique for Image Sequence Analysis," in *Proc. of Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pp. 13-20, Louvain-la-Neuve, Belgium, June 1997.

[8] Y. Avrithis, A. Doulamis, N. Doulamis and S. Kollias, "A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databases," *Computer Vision and Image Understanding*, Vol. 75, No. 1/2, Jul 1999, pp. 3-24.

[9] B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Videos," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 5, pp. 533- 544, Dec. 1995.

[10] K. S. Tang, K. F. Man, S. Kwong and Q. He, "Genetic Algorithms and Their Applications," *IEEE Signal Processing Magazine*, pp. 22-37, Nov. 1996.

[11] H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press, 1975.

[12] C. Gu, and M.-C. Lee, "Semiautomatic Segmentation and Tracking of Semantic Video Objects," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 572-584, Sept. 1998.

[13] A. D. Doulamis, N. D. Doulamis, K. S. Ntalianis and S. D. Kollias, "Unsupervised Semantic Object Segmentation of Stereoscopic Video Sequences," *Proc. of IEEE Int. Conf. on Intelligence, Information and Systems*, Washington D.C, USA, November 1999.

[14] O. J. Morris, M. J. Lee and A. G. Constantinides, "Graph Theory for Image Analysis: an Approach based on the Shortest Spanning Tree," *IEE Proceedings*, Vol. 133, pp.146-152, April 1986.

[15] A. Doulamis, N. Doulamis, K. Ntalianis, and S. Kollias, "Efficient Unsupervised Content-Based Segmentation in Stereoscopic Video Sequences," *Journal of Artificial Intelligence Tools, World Scientific Press*, vol. 9, no. 2, pp. 277-303, June 2000.

ARITHMETIC ENTROPY CODING FOR LOSSLESS WAVELET IMAGE COMPRESSION

G. A. Triantafyllidis and M. G. Strintzis

Information Processing Laboratory
Aristotle University of Thessaloniki
Thessaloniki 540 06, Greece
Tel. : +3031.996351, Fax : +3031.996342
Email: gatrian@iti.gr, strintzi@eng.auth.gr

ABSTRACT

Discrete wavelet transforms are widely used for lossless image compression. The overall performance of these schemes may be further improved by properly designing efficient entropy coders. A novel technique for the implementation of context-based adaptive arithmetic entropy coding is presented in this paper. This technique is based on the prediction of the value of the current transform coefficient, employing a weighted least squares method, in order to achieve appropriate context selection for arithmetic coding. Experimental results illustrate and evaluate the performance of the proposed technique.

1. INTRODUCTION

Two entropy coding methods are well-known and widely used: the Huffman and the arithmetic. The first method is preferable only when there is a lack of hardware resources and coding/decoding speed is a prime objective [1]. Arithmetic is somewhat slower than Huffman, but it is much more versatile and effective. In most cases, the adaptive variant of arithmetic coding is used [2],[3], in order to take advantage from high order dependencies with the use of conditioning contexts.

The arithmetic data compression technique encodes data by creating code string which represents a fractional value on the number line between 0 and 1. On each recursion of the algorithm only one symbol is encoded. The algorithm successively partitions an interval of the number line between 0 and 1, and retains one of the partitions as a new interval. Thus, the algorithm successively deals with smaller intervals, and the code string lies in each of the nested intervals.

This work was supported by the IST European Project HISCORE

The performance of arithmetic coders depends mainly on the estimation of the probability model which the coder will use. The coder can achieve an average output code length very close to the entropy corresponding to the probability model it utilises. Therefore, if the probability model accurately reflects the statistical properties of the input, arithmetic coding will approach the entropy of the source. Different probability models will give different compression performance for the same data. Thus, a scheme employing adaptive calculation of the probability will be better than a non-adaptive scheme, as it will allow a better approximation to the ”true” statistics of the data. The probabilities that an adaptive model assigns may change as each symbol is transmitted, based on the symbol frequencies seen so far in the message. A drawback of arithmetic coding of images using the above adaptive model is that it does not take into account the high amount of correlation between adjacent pixels. That is, each pixel is encoded using a probabilistic model adapted to all pixel values seen so far on the image. In this work, to alleviate this disadvantage a method similar to the one in [8] is adopted, with which, for every new coefficient to be encoded, the model is updated more than once, making the probabilistic model more adaptive to recent pixels, and thus more effective.

Every transform coefficient is put into one of several classes (buckets) depending on the weighted values of a set of previously entropy coded coefficients. To each context type corresponds a different probability model and thus each subband coefficient is compressed with an entropy coder following the appropriate model. The key issue is then how to find an efficient context based classification.

In our work, the Magnitude-Set Variable-Length-Integer representation (as proposed in [4]) is employed to represent the transform coefficients. According to this, every coefficient is classified into one of a set of ranges called mag-

nitude sets M, followed by the sign bit and the magnitude difference bits. For example, the numbers 15 and -16 are transmitted with the number triads $(7,+,3)$ and $(8,-,0)$ respectively.

This paper is organized as follows: In Section 2 the unweighted and the weighted least squares error methods are developed to determine the prediction of the current coefficient, in order to implement the context based adaptive arithmetic coding. Section 3 presents experimental results obtained when the proposed arithmetic entropy coder is applied, compared to S+P entropy coder. Finally, conclusions are drawn in Section 4.

2. PREDICTION AND CONTEXT SELECTION

2.1. Weight Optimization via Simple Linear Regression

The magnitude set M of the current pixel is estimated using the weighted values of coefficients that have already been entropy encoded in the current band, in the sister band(s) and in the parent band, in the pyramid structure, i.e., the predictor has the form:

$$\hat{M} = \sum_{i=0}^{N} a_i M_i \tag{1}$$

where the M_i indicates a previously encoded Magnitude Set, $\hat{M}$ is the prediction of the current Magnitude Set and the weights a_i, $1 \leq i \leq N$ are determined via linear regression so that $\hat{M}$ are least squares estimates of M.

Experimental results have proved that the magnitude sets of the coefficients shown in Figure 1, which differ in shape for every subband LH, HL and HH, suffice for an accurate prediction of the magnitude set M of the current pixel. This scheme implies that the subbands will be coded in the following order: first the LH band, then the HL band and finally the HH band, so that to establish the necessary casual relationship. Therefore, Eq. (1) can be expressed for each of the subbands as follows [12]:

$$\begin{aligned}
\underline{LH\ band:}\quad & \hat{M} = a_1 M_w + a_2 M_{nw} + a_3 M_n + a_4 M_{ne} \\
& + a_5 M_{p1} + a_6 M_{p2} \\
\underline{HL\ band:}\quad & \hat{M} = a_1 M_w + a_2 M_{nw} + a_3 M_n + a_4 M_{ne} \\
& + a_5 M_{p1} + a_6 M_{p2} + a_7 M_{sis} \\
\underline{HH\ band:}\quad & \hat{M} = a_1 M_w + a_2 M_{nw} + a_3 M_n + a_4 M_{ne} \\
& + a_5 M_{p1} + a_6 M_{p2} + a_7 M_{sis1} + a_8 M_{sis2}
\end{aligned} \tag{2}$$

Subscripts w, nw, n, ne are directional short notations for west, north-west, north and north-east respectively, p_k ($k = 1, 2$) indicates the k^{th} parent pixel and sis indicate the corresponding pixels or pixel in the sister bands. Using further coefficients for the estimation of the magnitude set of the current coefficient would cause extra computational load which is not justified by the improvement of the results.

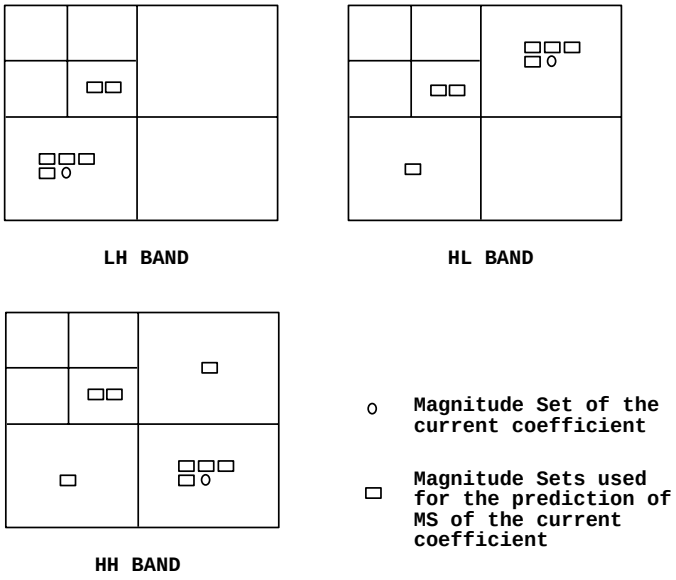


Fig. 1. Pixels employed for the prediction of the magnitude set of current coefficient for each subband

Let the matrix $\mathbf{S}$ have $N \times N$ rows (where $N \times N$ are the dimensions of the image to be coded) and eight columns. Each row consists of all the previously encoded magnitude sets used for the estimation of the current magnitude set, i.e., the subscript of each element of the matrix indicates the current coefficient and the superscript indicates one of the eight previously encoded magnitude sets used for the estimation of the current coefficient. Further, we form a vector $\mathbf{y}$ composed of all magnitude sets, i.e., the subscript of each element of this vector indicates the current coefficient.

$$\mathbf{S}=\begin{bmatrix} M_0^1 & M_0^2 & M_0^3 & \dots & M_0^8 \\ M_1^1 & M_1^2 & M_1^3 & \dots & M_1^8 \\ M_2^1 & M_2^2 & M_2^3 & \dots & M_2^8 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M_{N\times N}^1 & M_{N\times N}^2 & M_{N\times N}^3 & \dots & M_{N\times N}^8 \end{bmatrix}$$

$$\mathbf{y}=\begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{N\times N} \end{bmatrix} \tag{3}$$

Then, the vector $\mathbf{a}$ of the optimal weights can be formed as [10]:

$$\mathbf{a} = (\mathbf{S}^T\mathbf{W}\mathbf{S})^{-1}\mathbf{S}^T\mathbf{W}\mathbf{y} \tag{4}$$

where $\mathbf{W}$ is the weighted linear regression matrix which may be chosen to be either the unity matrix ([6], unweighted linear regression) or a user-defined appropriate weighted matrix ([12], weighted linear regression).

Having calculated the weights, the norms of (2) can be used to classify the current transform coefficient to the proper bucket, that is to determine which probabilistic model to use during the adaptive arithmetic entropy coding.

In order to encode the LL band it can be decomposed again and similar techniques outlined above can be used to encode the high pass bands at this second level of decomposition. This procedure can be carried on until the low pass band is of very small size and can be transmitted in an uncoded manner.

2.2. Weight Optimization via appropriate Weighted Linear Regression

In this case, instead of using a simple unweighted linear regression algorithm, a more sophisticated method is implemented to find the best weights for the estimation of the current coefficient. Experiments have shown that the larger errors in estimating the transformed coefficients occur on the edges of the transformed image. As a result, the most appropriate matrix $\mathbf{W}$ of equation (4) must have higher weights in the positions which correspond to the edges of the transformed image. A Canny edge detector operator [9] is employed for this task.

Having calculated the appropriate weighted matrix $\mathbf{W}$, equation (4) is used for the calculation of vector $\mathbf{a}$. Then, the norms of (2) are employed to estimate the magnitude set of the current coefficient.

3. EXPERIMENTAL RESULTS

The above context-based arithmetic entropy coding technique was compared to the method used in the widely regarded as state-of-the-art algorithm of S+P [4]. Our experiments may be summarized as follows:

Step 1 Apply the S+P transform for the initial decorrelation of the selected image.

Step 2 Apply the algorithm of unweighted or weighted least squares method for the prediction of the magnitude sets.

Step 3 Classify the magnitude sets of each coefficient into one of several buckets depending on the weighted values of the selected set of previously entropy coded coefficients.

Step 4 Apply adaptive arithmetic entropy coding [2] to each bucket. Aiming to better adaptivity, for every new coefficient to be encoded, the model is updated three times instead of once. The sign and magnitude difference are also arithmetically coded but using a fixed (instead of adaptive) uniform distribution model, in order to increase the computational efficiency.

The arithmetic entropy coder proposed, was applied to standard black and white, 8 bpp, images following an S+P transform [4]. Table 1 presents the results of the unweighted or weighted least squares methods compared to that of the S+P entropy coder.

The computational speed of the proposed method is somewhat slower than the simple arithmetic entropy coding since for each image to be encoded, the calculation of the weights has to be performed in order to conclude to the fittest possible weights. However, if we want, we can go one step further: to use the linear regression algorithm for the weights calculation with a whole set of typical images and find a fixed set of weights. In that case, it is clear that compression performance for each image is going to be decreased but the overall speed of the algorithm will be improved.

image	S+P		method I		method II	
	bytes	bpp	bytes	bpp	bytes	bpp
lena	136702	4.17	136007	4.15	135724	4.14
peppers	150182	4.58	149363	4.56	148865	4.54
crowd	131010	4.00	130326	3.98	130123	3.97
boat	141272	4.31	139917	4.27	139591	4.26
airplane	128238	3.91	126985	3.88	126838	3.87
bridges	182882	5.58	182192	5.56	181198	5.53
harbour	154896	4.73	153141	4.67	152207	4.64
barbara	149254	4.55	147861	4.51	147108	4.49

Table 1. Number of bytes and bits per pixel needed for entropy coding with optimal weights calculated via linear regression with matrix $\mathbf{W} = \mathbf{I}$ (method I) or $\mathbf{W}$ weighted (method II) compared to S+P entropy coding.

4. CONCLUSIONS

A method was presented for the implementation of an efficient context-based arithmetic entropy coding. The method employs unweighted or weighted least squares techniques to determine the weights used so as to estimate the magnitude set of the current coefficient, based on a selected set of magnitude sets of pixels which have been previously coded. Experiments show that the use of the weighted least squares algorithm leads to better results, and consistently outperforms the entropy coder proposed by the state-of-the-art method S+P in [4].

5. REFERENCES

[1] M. Rabbani and P.W. Jones, "Digital Image Compression Techniques", Bellingham, WA: SPIE, 1991.

[2] R. N. Williams, "Adaptive Data Compression", Norwell, MA: Kluwer, 1991.

[3] R. M. Witten, I. H. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression", Communications of the ACM, vol. 30, pp 520-540, June 1987.

[4] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy compression", IEEE Transactions on Image Processing, vol. 5, pp 1303-1310, Sept. 1996.

[5] Y. Yoo, A. Ortega, and B. Yu, "Adaptive Quantization of Image Subbands with Efficient Overhead Rate Selection", Proc. of the Intl. Conf. On Image Proc., ICIP96, vol. 2, (Lausanne, Switzerland), pp 361-364, Sept. 1996.

[6] N. Memon, X. Wu, and B.-L. Yeo, "Entropy Coding Techniques for Lossless Image Compression with Reversible Integer Wavelet Transforms", IBM Research Report, Computer Science/Mathematics, RC 21010, Oct. 22, 1997.

[7] X. Wu and N. D. Memon, "Context-Based Adaptive Lossless Image Coding", IEEE Transactions on Communications, vol. 45, pp 437-444, Apr. 1997.

[8] X. Wu, K. U. Barthel and G. Ruhl, "Adaptation to Nonstationarity of Embedded Wavelet Code Stream", Proc. of the Intl. Conf On Image Proc., ICIP98, (Chicago, IL), Oct. 1998.

[9] J. Canny, "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp 679-698, Nov. 1986.

[10] J. A. Cadzow, "Signal Processing via Least Squares Error Modeling", IEEE Magazine of ASSP, pp 12-31, Oct. 1990.

[11] S. Efstratiadis, D. Tzovaras, and M. G. Strintzis, "Hierarchical Partition Priority Wavelet Image Compression", IEEE Transactions on Image Processing, Vol. 5, No. 7, pp 1111-1123, July 1996.

[12] G. A. Triantafyllidis and M. G. Strintzis, "A Context Based Adaptive Arithmetic Coding Technique for Lossless Image Compression", IEEE Signal Processing Letters, Vol. 6, No. 7, pp 168-170, July 1999.

Content-based Watermarking for Indexing Using Robust Segmentation *

Nikolaos V. Boulgouris, Ioannis Kompatsiaris, Vasileios Mezaris, and Michael G. Strintzis
Information Processing Laboratory
Electrical and Computer Engineering Dept.
Aristotle University of Thessaloniki
54006 Thessaloniki, GREECE
e-mail: strintzi@dion.ee.auth.gr

ABSTRACT

In this paper, a novel approach to image indexing is presented using content-based watermarking. Some concepts associated with the application of watermarking to image indexing are discussed and a segmentation algorithm, appropriate for content-based watermarking, is presented. The segmentation algorithm is applied on reduced images and derives the exact same objects when performed on either the original or the watermarked image. In this way, the proposed system does not suffer from synchronization problems that usually occur during watermark detection.

1 Introduction

Watermarking has received significant attention lately due to its applications on the protection of intellectual property rights (IPR) [1, 2]. However, many other applications can be conceived which involve information hiding [3]. In this paper, we propose the employment of watermarking as a means to perform content-based indexing and retrieval of images from data bases.

In order to endow the proposed scheme with content-based functionalities, information must be hidden region-wise in digital images. Thus, the success of any content-based approach depends largely on the segmentation of the image based on its content. In the present paper, an efficient segmentation algorithm is used prior to information embedding.

Embedding indexing information in image objects has the following advantages:

- Each region in the image carries its own description and no additional information must be kept for its description.
- The image can be moved from a database to another without the need to move any associated description.

*This work was supported by the EU CEC Project ASPIS and the Greek Secretariat of Research and Technology Project PENED99. The help of COST211quat is also gratefully acknowledged.

The present paper provides an elegant framework for the application of watermarking to image indexing. Despite the fact that, due to the increased computational complexity, the proposed system cannot be used in large data bases at the present time, it clearly demonstrates another potential application of watermarking.

The paper is organized as follows: the system overview is given in section 2. The information embedding method is presented in section 3. In section 4, the segmentation algorithm used with our system is described. In section 5, experimental evaluation is shown, and finally conclusions are drawn in section 6.

2 System overview

The methodology proposed in this paper assumes that the watermarked images will not be subjected to any attacks. The investigation of robust watermarking or copyright protection is beyond the scope of this paper. However, even without attacks, watermark detection is not a straightforward procedure in our content-based framework since the objects of the images, rather than the original rectangular images, are watermarked (see fig. 1). Thus, the first step in the watermark detection process is to segment the image into objects and subsequently, based on this segmentation, to extract the information bits associated with each object (see fig. 2). If during detection, the image is segmented in different regions than those used in the embedding, then

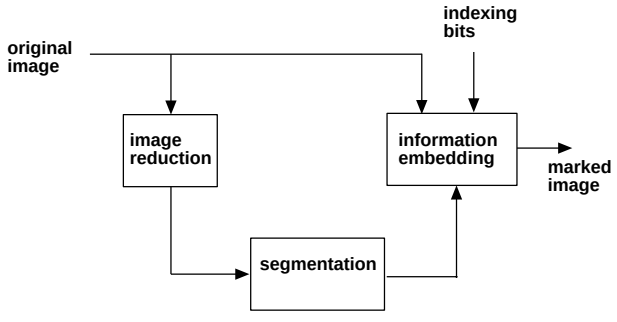


Figure 1: Block diagram of the embedding scheme.

the detection process will not be synchronized with the embedding process and the embedded information will not be retrieved.

In order to combat the loss of synchronization, special care should be taken to make sure that the execution path followed during the embedding process for segmenting the image into objects and the execution path during detection will be identical despite the alteration the image has undergone due to watemarking.

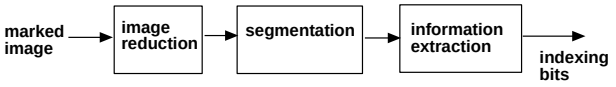


Figure 2: Block diagram of the detection scheme.

For this to be achieved, the algorithm is applied to a reduced image, comprising of the mean values of the pixel intensities in 8×8 pixel blocks of the original image. Thus, if the mean values of the intensities for each such block remain the same after the watermarking process, then the resulting reduced image, to which the segmentation algorithm is applied, is the same regardless of whether it is derived from the original or the watermarked image.

A space-domain information embedding strategy which keeps the mean value of each image block constant is described in the ensuing section.

3 Information embedding

The mean intensity $\bar{I}[l_1, l_2]$ of an 8×8 block in the original image I is

$$\bar{I}[l_1, l_2] = \frac{1}{64} \sum_{i=0}^{7} \sum_{j=0}^{7} I[8l_1 + i, 8l_2 + j]$$

for all three colour components, where l_1, l_2 are the block indexes. We choose to embed the watermark bits in the blue component of the RGB images because the Human Visual System is less sensitive to blue colours [4]. One bit is embedded in each 8×8 block.

The intensities $\bar{I}'[l_1, l_2]$ of the reduced image that will be derived from the watermarked image during detection, are:

$$\bar{I}'_R[l_1, l_2] = \frac{1}{64} \sum_{i=0}^{7} \sum_{j=0}^{7} I_R[8l_1 + i, 8l_2 + j] = \bar{I}_R[l_1, l_2]$$

$$\bar{I}'_G[l_1, l_2] = \frac{1}{64} \sum_{i=0}^{7} \sum_{j=0}^{7} I_G[8l_1 + i, 8l_2 + j] = \bar{I}_G[l_1, l_2]$$

$$\bar{I}'_B[l_1, l_2] =$$

$$\frac{1}{64} \sum_{i=0}^{7} \sum_{j=0}^{7} (I_B[8l_1 + i, 8l_2 + j] + b \cdot a_{l_1,l_2}[i, j] \cdot w[i, j]) \quad (1)$$

where b is the embedded bit (valued -1,1), a is the watermark strength factor, which depends on a local energy measure and is computed seperately for each block in the original image, and $w[i, j]$ is the watermark matrix (see fig. 3) given by

$$w[i, j] = \begin{cases} 1, & \text{if } i + j = \text{even} \\ -1, & \text{if } i + j = \text{odd} \end{cases} \quad (2)$$

1	-1	1	-1	1	-1	1	-1
-1	1	-1	1	-1	1	-1	1
1	-1	1	-1	1	-1	1	-1
-1	1	-1	1	-1	1	-1	1
1	-1	1	-1	1	-1	1	-1
-1	1	-1	1	-1	1	-1	1
1	-1	1	-1	1	-1	1	-1
-1	1	-1	1	-1	1	-1	1

Figure 3: Watermark matrix. The information bit is multiplied by the matrix elements and the strength factor. The resulting signal is added on the image.

The watermark strength a is varied according to the intensity differences of the blue component in the area of the current pixel, as follows:

$$a_{l_1,l_2}[i, j] = \begin{cases} 3, & \text{if } 2T < D_i + D_j \\ 2, & \text{if } T < D_i + D_j \leq 2T \\ 1, & \text{otherwise} \end{cases}$$

where

$$D_i = |I_B[8l_1 + i - 1, 8l_2 + j] - I_B[8l_1 + i + 1, 8l_2 + j]|$$

$$D_j = |I_B[8l_1 + i, 8l_2 + j - 1] - I_B[8l_1 + i, 8l_2 + j + 1]|$$

and the threshold T was experimentally set to 10.

Equation (1) yields

$$\bar{I}'_B[l_1, l_2] = \frac{1}{64} \sum_{i,j} I_B[8l_1 + i, 8l_2 + j] + \frac{1}{64} \sum_{i,j} (b \cdot a_{l_1,l_2}[i, j] \cdot w[i, j]) =$$

$$= \bar{I}_B[l_1, l_2] + \frac{b}{64} \sum_{i+j:even} (a_{l_1,l_2}[i, j] \cdot 1) + \frac{b}{64} \sum_{i+j:odd} (a_{l_1,l_2}[i, j] \cdot (-1)) =$$

$$= \bar{I}_B[l_1, l_2] + \frac{b}{64}\Big(\sum_{i+j:even} a_{l_1,l_2}[i,j] - \sum_{i+j:odd} a_{l_1,l_2}[i,j]\Big) \quad (3)$$

In order to make sure that the application of the segmentation algorithm to either the original or the watermarked image produces the same results, the mean block intensities of the block (l_1, l_2) should be equal, i.e

$$\bar{I}'_B[l_1, l_2] = \bar{I}_B[l_1, l_2]$$

From (3), it is seen that the above equation is equivalent to

$$\sum_{i+j:even} a_{l_1,l_2}[i,j] = \sum_{i+j:odd} a_{l_1,l_2}[i,j] \quad (4)$$

Thus, the equality of the mean block intensities can be achieved by appropriately modifying the watermark strength factor $a_{l_1,l_2}[i,j]$, after it has been calculated, so that condition (4) is met. In our scheme this is done by reducing the values of $a_{l_1,l_2}[i,j]$, for $i+j : even$ or $i+j : odd$, depending on which sum is greater.

4 Segmentation algrithm

Segmentation methods for 2D images may be divided primarily into region-based and boundary-based methods. Region-based approaches [5] rely on the homogeneity of spatially localised features such as gray level intensity and texture. Region-growing and split and merge techniques also belong to the same category. On the other hand, boundary-based methods use primarily gradient information to locate object boundaries. In this paper, a region-based approach is taken. The image segmentation algorithm consists of three stages:

1. An initial estimation of the number of a connected regions and their colour and spatial centers.
2. An iterative pixel classification process that uses the initial estimations as a starting point.
3. A feature extraction process for indexing purposes.

For notational simplicity, the reduced image to which the segmentation algorithm is applied will be heareafter denoted as I. An initial estimation of the number of classes contained in the (reduced) image is produced by dividing the image into N non-overlapping blocks, each represented by the mean values of the three colour coordinates of the CIE L*a*b* colour space [6], $\bar{\mathbf{I}}_n = (\bar{I}_{L,n}, \bar{I}_{a,n}, \bar{I}_{b,n})$, $n = 1, \ldots, N$, and applying the max-imin algorithm to the set of blocks. The colour distance between two blocks is defined as:

$$\|\bar{\mathbf{I}}_m - \bar{\mathbf{I}}_n\| = \sqrt{(\bar{I}_{L,m} - \bar{I}_{L,n})^2 + (\bar{I}_{a,m} - \bar{I}_{a,n})^2 + (\bar{I}_{b,m} - \bar{I}_{b,n})^2}$$

A simple K-means algorithm [7] is then used to classify each block to one of the classes; those classes that do not form connected regions are split, so that K connected regions s_k are formed; the mean values of the colour and space coordinates over the set of blocks of each region constitute an initial estimation of the colour centers $\bar{\mathbf{I}}_k = (\bar{I}_{L,k}, \bar{I}_{a,k}, \bar{I}_{b,k})$, and spatial centers $\bar{\mathbf{S}}_k = (\bar{S}_{x,k}, \bar{S}_{y,k})$, $k = 1, \ldots, K$. These centers are used as a starting point by the pixel classification algorithm: a K-Means-with-connectivity-constraint algorithm. The distance of a pixel $p = (p_x, p_y)$ from a region s_k is defined as follows:

$$D(\mathbf{p}, k) = \|\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}_k\| + \frac{\lambda}{A_k}\|\mathbf{p} - \bar{\mathbf{S}}_k\|$$

where $\|\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}_k\|$ is the colour distance:

$$\|\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}_k\| = \sqrt{(I_L - \bar{I}_{L,k})^2 + (I_a - \bar{I}_{a,k})^2 + (I_b - \bar{I}_{b,k})^2}$$

$\|\mathbf{p} - \bar{\mathbf{S}}_k\|$ is the spatial distance:

$$\|\mathbf{p} - \bar{\mathbf{S}}_k\| = \sqrt{(p_x - \bar{S}_{x,k})^2 + (p_y - \bar{S}_{y,k})^2}$$

A_k is the area of the region s_k, defined as $A_k = M_k$, where M_k is the number of pixels assigned to s_k, and λ is a regularisation parameter. The K-Means-with-connectivity-constraint algorithm features region splitting of non-connected regions and merging of chromatically similar neighboring regions. The region centers are recalculated on every iteration as the mean values of the colour and space coordinates over the set of pixels assigned to each region. If M_k elements are assigned to s_k then

$$\bar{\mathbf{I}}_k = \frac{1}{M_k}\sum_{m=1}^{M_k} \mathbf{I}(\mathbf{p}_m^k)\ , \quad (5)$$

$$\bar{S}_k = \frac{1}{M_k}\sum_{m=1}^{M_k} p_m^k, \quad (6)$$

where $\mathbf{p}^k = (p_x^k, p_y^k)$ are the pixels assigned to region s_k. The centers corresponding to regions that fall below a size threshold are omitted. As soon as the K-means-with-connectivity-constraint algorithm converges, regions smaller than a specified threshold are appended to other regions, to ensure that no particularly small, meaningless regions are formed. Finally, a set of features is extracted for every region; those include colour, position, size, and shape features.

5 Experimental results

The watermarking and segmentation algorithms described in the previous sections were tested for embedding information in a variety of colour test images (Fig.

4). 256 bits were embedded on each image object. The embedded information was the values of the features used by the ISTORAMA[1] content-based image retrieval system. Alternatively, any other kind of object-related information could be embedded, including a short text describing the object. The bits were embedded in the blue component of RGB images using the procedure described in section 3. No perceptual degradation of image quality was observed.

Almost all embedded bits could be reliably extracted from the watermarked image. Only a tiny portion ($\leq 0.1\%$) of the total embedded bits were extracted in error. For this reason, simple channel coding was used to ensure errorless information extraction.

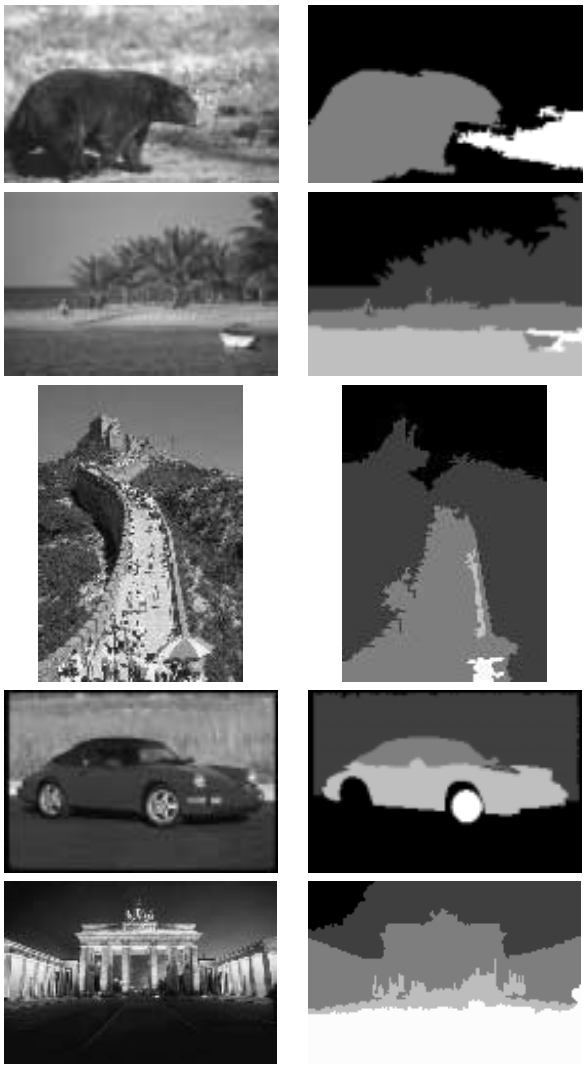


Figure 4: Images segmented into regions.

[1] http://uranus.ee.auth.gr/Istorama

6 Conclusions

A methodology was presented for the content-based embedding of indexing information in digital images. Watermark information is embedded after images are segmented into objects. The watermarking algorithm does not change the mean value of image blocks so that the same objects are extracted during embedding and detection.

The proposed system is appropriate for retrieving images from small data bases only, since the need for segmentation at the detector may be computationally intensive, delaying the process. However, it demonstrates a potentially useful application of watermarking.

7 References

[1] W. Zeng and B. Liu, "A Statistical Watermark Detection Technique Without Using Original Images for Resolving Rightful Ownerships of Digital Images," *IEEE Trans. Image Processing*, vol. 8, no. 11, pp. 1534–1548, Nov. 1999.

[2] D. Simitopoulos, N. V. Boulgouris, A. Leontaris, and M. G. Strintzis, "Scalable detection of perceptual watermarks in JPEG2000 images," in *Proc. CMS 2001*, May 2001.

[3] A. M. Alattar, "Smart Images Using Digimarc's Watemarking Technology," in *Proc. SPIE*, Jan 2000.

[4] M. Kutter, F. Jordan, and F. Bossen, "Digital Signature of Color Images Using Amplitude Modulation," in *Proc. SPIE*, Feb 1997.

[5] P. Salembier and F. Marques, "Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1147–1169, December 1999.

[6] S. Liapis, E. Sifakis, and G. Tziritas, "Color and/or Texture Segmentation using Deterministic Relaxation and Fast Marching Algorithms," in *Intern. Conf. on Pattern Recognition*, September 2000, vol. 3, pp. 621–624.

[7] J. McQueen, "Some Methods for Classification and Analyis of Multivariate Observations," in *5th Berkely Symp. on Math. Stat. and Prob.*, 1967, vol. 1, pp. 281–296.

Query by Image Content using NOKIA 9210 Communicator

Ahmad Iftikhar[1], Faouzi Alaya Cheikh[2], Bogdan Cramariuc[2] and Moncef Gabbouj[2]

Abstract—In this paper we present a new Java-based client-server application for content-based image retrieval over wireless networks. The application on the client side is running on the NOKIA's 9210 Communicator and is written in pure Java™.

Index Terms— Content, Retrieval, Indexing, Multimedia, Image, Search, Wireless, Mobile, Communicator.

I. INTRODUCTION

A. Wireless Communications and Terminals

The way people are communicating is changing very fast. Few years ago, mobile phones were lucrative items restricted to a very small community of rich businessman and government agents. Moreover, they were used exclusively for voice calls.

Today the mobile terminal penetration is growing steadily and continuously. And their use is no longer restricted to voice communication only. In Finland, it is widely accepted among youngsters to use a GSM phone for sending SMS messages, to chat with friends or to play games. Adults may be more interested in checking their stocks or paying a bill using their wireless terminal and the Wireless Application Protocol (WAP). In Japan a phenomenal change in the use of mobile phones happened by the introduction of the "iMode" [IMODE] system. The number of users since its introduction two years ago has risen to 17 millions.

The third generation, or 3G [3G], phones will create new opportunities for content providers, by providing a way of transmitting text, voice, images, and streamed video. Moreover, their ability to be connected to the Internet all the time will provide users with an overwhelming access to a huge amount of information. Users will then face the problem of how to retrieve the information of interest to them in an efficient manner. The goal is to allow for searching and navigation in this wealth of data without the need to make text-based queries for three obvious reasons:

- The user may be unable to type in commands.
- The keyboards of portable devices are not very comfortable for text-based commands.
- Text-based queries may not be very appropriate in the case of images, video or music.

Therefore, a content-based indexing and retrieval engine coupled to a speech recognition engine could be the ultimate interface to such a system

In this paper we will introduce a content-based search engine and its graphical user interface. A demo of the system will be given during the presentation. The speech recognition part is not considered in this paper.

Even though, the newly introduced pervasive devices are having faster processors, larger memories and their available communication bandwidth is getting wider, they remain far behind the PC capabilities. Therefore, a major challenge in designing such a system is to understand the characteristics of such devices and their hardware and software limitations.

B. Content-based Indexing and Retrieval

Since the early 1990s, content-based indexing and retrieval (CBIR) of digital images became a very active area of research. Both industrial and academic systems for image retrieval have been built. Most of these systems (e.g. *QBIC*™ [QBIC] from IBM, *NETRA* [NETRA] from UCSB, *Virage* [VIRAGE] from Virage Inc., *MUVIS* [MUVIS] from TUT) support one or more of the following options: browse, search by example, search based on a single or a combination of low level features. These features can be extracted from the image, such as color, shape, texture, spatial layout of objects in the scene or added to it after its capture, such as contextual information and keywords

II. CLIENT-SERVER ARCHITECTURE

A. The Client Side: The Nokia 9210 Communicator

1) Introduction

Nokia 9210 Communicator [NOKIA] is a major step forward in the road to the Mobile Internet environment. This pioneering product showcases the key elements in future mobile communications, such as easy navigation and input, a high-quality color display, mobile messaging with high data speed, imaging and video clips. Additionally, Java support and Symbian's OS (operating system) [EPOC]

[1] Nokia Mobile Phones, P.O. Box 1000 (Visiokatu 3, Tampere 33720), FIN-33721 Tampere, Finland

[2] Tampere University of Technology, P.O. Box 553, FIN-33101, Tampere, Finland

bring open development interfaces to the Nokia 9210 Communicator for numerous additional applications to be provided by any third party developers.

2) Hardware Details

Nokia 9210 communicator [9210F] contains 32-bit ARM-based RISC processor. It has 8 MB (SD-RAM) of execution memory and its C drive (serial flash) is of 4MB. It can have a multimedia card of up to 64MB, see Figure 1.

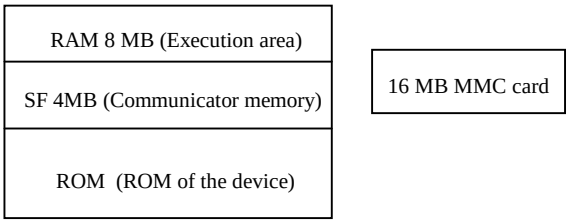


Figure 1: Memory configuration of the Nokia 9210

It has a color display of 4096 colors. Display size is 640 x 200 pixels. In addition, it has a relatively large size keyboard and is capable of making high speed data calls. It uses Symbian's operating system Crystal 6.0.

Java virtual machine consumes about 2.1MB. The proposed implementation of the Java application consumes approximately 397KB of memory.

3) Operating System

Symbian's platform [SYMB] is a robust, object oriented operating system for devices with limited capabilities (small memory, little computing power, sensitive to power consumption). Devices using this system do not need to reboot often as this OS is stable, does not leak memory (or very little) and manages the system resources efficiently.

Since Symbian's platform devices have little memory, small secondary storage and less computational power, applications written for this systems must be efficient. This is especially the case of Symbian Crystal release 6.0 [CRYST] intended for wireless media.

The proposed image search engine deals with images and thus consumes a large amount of memory. The system must thus be well managed to avoid such memory related problems. A high-speed data link is used. Actually the 9210 supports data links up to 43.2KB (High Speed Circuit Switched Data, HSCSD) [9210F], but we are using 38.4KB. High-speed data call reduces airtime but it is a costly option. Airtime will be reduced in 3G systems, and thus queries can be made without using a high-speed data call. Only a high-speed connection for data transfer will be needed.

Personal Java [PJAVA] is ported on the Crystal 6.0 that is compatible with JDK1.1.8 [JDK]. But current implementation of Personal Java is not supporting swing (a pure graphics APIs of Java). Personal Java consumes 2.1 MB of memory when just VM is up (without Java application). The Java Application (image search engine) takes an additional 397 KB of RAM leaving very little memory for images. In this implementation, images are fetched when requested to be displayed and discarded when not need.

B. The Server Side

On the server side we are using a Servlet [SERV]. Servlets are Java programs that extend the capabilities of the server. They are similar to applets in a browser. The client sends the query to the servlet; which checks the query media type and passes it to the appropriate query handler.

The heavy processing required for the feature extraction, similarity estimation and results presentation are done on the server through calls from the Java side to methods implemented in native code. In this way we take advantage of the more efficient native code as compared to the pure Java implementation.

C. Communication Protocol

A communication protocol is defined between the client and the server. This Protocol specifies the media type (Image media, Video media, or Audio media, currently we are using Image media only), query type (random query from database, query by image data or query with an image from the database) and query data (image data if the image is not in the database or images' index in the database or image location URL).

The server sends back to the client the status of query execution and the results of the query, which consists of the list of names of the images and their similarity scores with respect to the query image. The client later fetches scaled versions (80 x 60 pixels) of the images to be presented to the user (in our experiments we requested 10 images). Scaling is done on the server side, in order to reduce the traffic. Only on the request of the user the full size image is fetched from the server.

III. THE USER INTERFACE AND SCREEN SIZE CONSIDERATION

As mentioned earlier, wireless devices have limited resources. In this application and in addition to the processing and memory issues, the designer has to consider the screen size of the wireless device. As the 9210 belongs to the communicator class, it has a relatively larger screen (640 x 200) [9210F]. When displaying the query results, images have been resized to fit the available display. As can be seen in the examples given in Figures 2-5, the image content can still be legible. Furthermore, in the 9210 we take advantage of the command button area and place the four most used commands there. The other commands are placed in the menu. The menu is displayed only when the user presses the menu button, and hence it is not consuming screen space when it is not active.

IV. Results and Assessments

Figure 2 shows the GUI and the menu toolbar on the Nokia 9210 Communicator which has been implemented for the proposed content-based image indexing and retrieval system. As can be seen, the screen space is fully utilized and the important features are displayed. Remember that the menu is not accessed very often, and thus the space is used to display the query results. The most commonly used buttons are assigned to the command button area on the right side of the screen.

Figures 3-5 show the results of different types of queries made to the image database, namely, color histogram, shape and texture queries. In each case, the top ten similar images are retrieved and displayed on the Communicator screen.

Due to the limitations imposed by the wireless device, the operating system and the communication channel, a rather slow query response has been achieved. Table 1 below shows the timing obtained in different queries made with the Nokia 9210 Communicator. Query time on Server starts when a query arrives at the servlet, the servlet extracts the image features and makes the query in the native code. It includes also the time to save the results on the server and creates a Java result object and passes it to the servlet to send it to the client. Sending result object to the client is the time to send Java object containing the image names and similarity scores for 50 images. The image retrieval time starts when the server passes the image retrieval request to the servlet. It includes the time needed to retrieve the actual images from the server' s file system and resize them on the server side. Finally, the image transfer time is the time it takes the server to transmit the resized images to the client.

Query type	Shape	Histogram	Texture
Query time on server (sec)	26	17	16
Sending result object (ms)	100	110	108
Image retrieval (ms)	230	245	260
Image transfer time (ms)	430	470	490

Table 1: Timing Results for Image Query

The timing provided in Table 1 should be interpreted with care. They varied quite a lot during the testing as they depend on a number of rather dynamic factors, such as the network traffic (load), the load on the server (as well as the state of the server, i.e., servlet must be uploaded again in case no one requested its use from the server) and the available memory on the device.

V. Conclusions and Future Work

A novel implementation of TUT' s MUVIS image query system has been proposed and tested on the new Nokia 9210 Communicator using a Java-based client server paradigm. A functional GUI was implemented taking into account the small size of the Communicator. The Demo shows that such an implementation is feasible; however, due to the limiting factors in both the hardware and software of the wireless terminal as well as the communication channel, very limited results have been obtained, namely, reduced sizes of image query results, small number of images, long process and access times. The good news is that with the advent of 3G networks, offering higher data rates and more processing power in wireless devices and more memory, such an application would be possible. Furthermore, a more efficient Java implementation, called J2ME (JAVA 2 Micro Edition) [J2ME], is under development. This implementation is targeted, among other applications, to small size wireless devices with limited capabilities.

VI. Acknowledgements

We gratefully acknowledge the support of Mr. Timo Ulmanen, from Nokia Mobile Phones for his efforts and support on behalf of this work.

VII. References

[IMODE] http://www.ntt.docomo.com/i/

[3G] http://www.3gpp.org/

[QBIC] http://wwwqbic.almaden.ibm.com/~qbic/

[NETRA] http://maya.ece.ucsb.edu/Netra/

[VIRAGE] http://www.virage.com/

[MUVIS] M.Trimeche, F.Alaya Cheikh, M.Gabbouj and Bogdan Cramariuc, "Content-based Description of Images for Retrieval in Large Databases:MUVIS," X European Signal Processing Conference, Eusipco-2000, Tampere, Finland, September 5-8, 2000, pp. 139-142.

[J2ME] http://java.sun.com/j2me/

[PJAVA] http://java.sun.com/products/personaljava/

[JDK] http://java.sun.com/products/jdk/1.1/

[SERV] http://java.sun.com/products/servlets/

[NOKIA] http://www.nokia.com/phones/9210/index.html

[9210F] http://www.nokia.com/phones/9210/features.html

[SYMB] http://www.symbian.com/

[CRYST] http://www.symbian.com/technology/v6-papers/v6-papers.html

[EPOC] http://www.epocworld.com

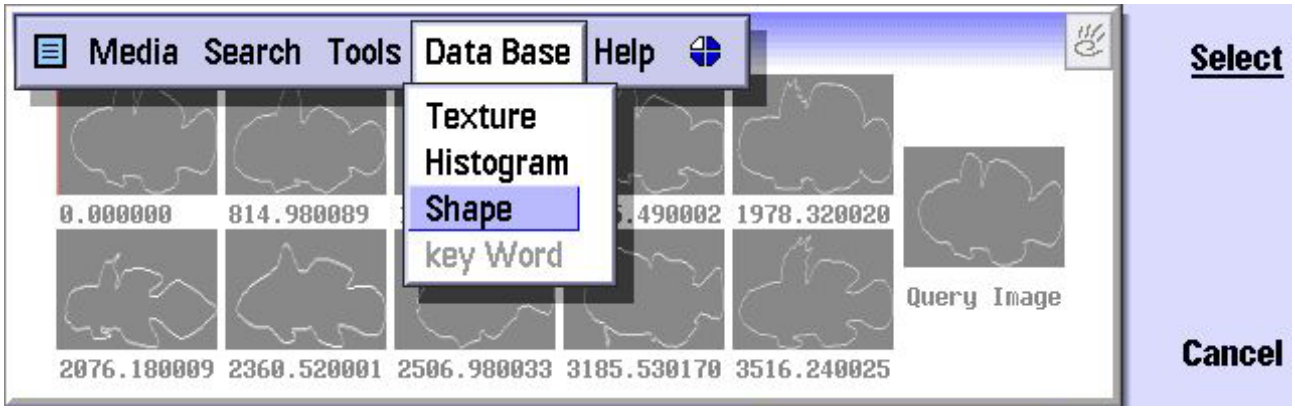


Figure 2: GUI on the Nokia 9210 Communicator

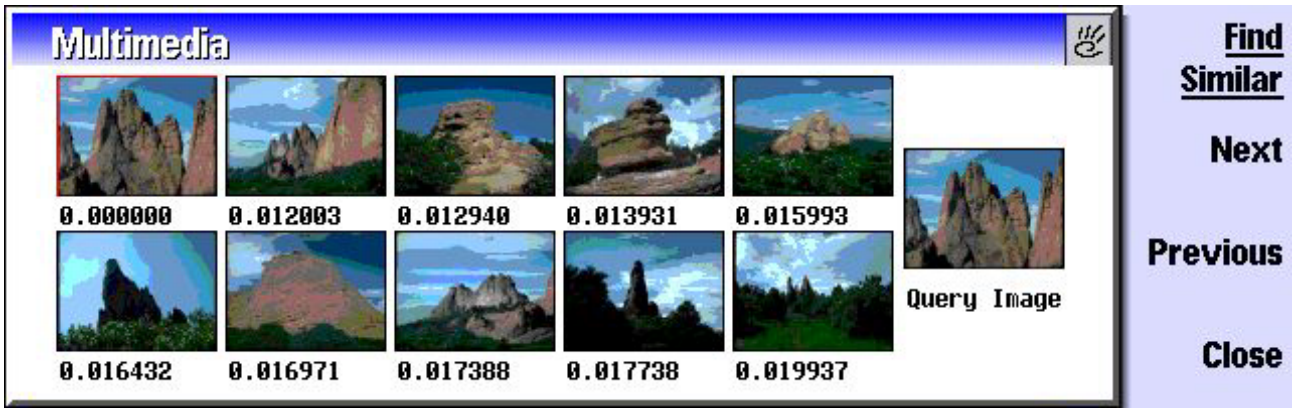


Figure 3: Results of a color histogram-based image query

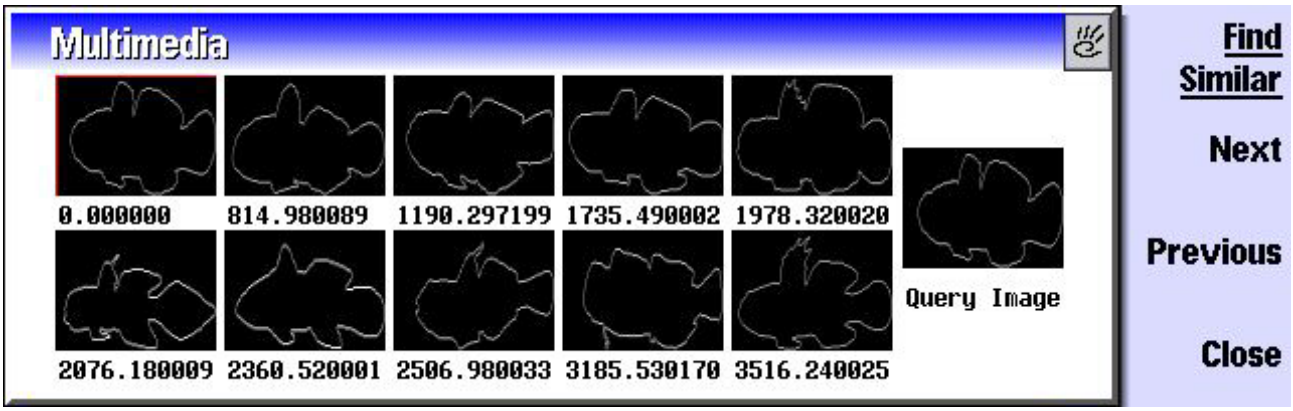


Figure 4: Results of a shape-based image query

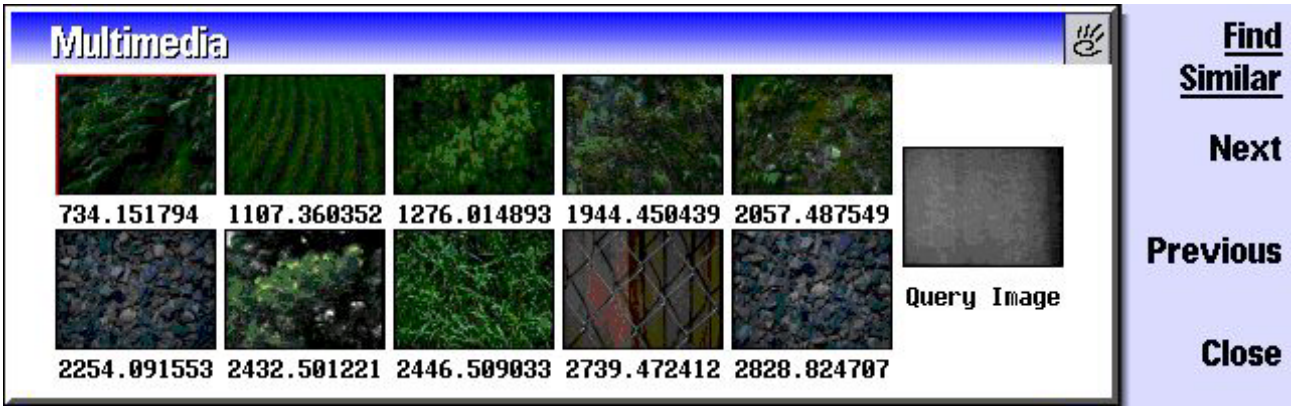


Figure 5: Results of a texture-based image query

Shape Similarity Estimation using Ordinal Measures

Faouzi Alaya Cheikh, Bogdan Cramariuc, Mari Partio, Pasi Reijonen and Moncef Gabbouj

Tampere University of Technology,
P.O. Box 553, FIN-33101, Tampere, Finland

Abstract— In this paper we present a novel approach to shape similarity estimation based on ordinal correlation. The proposed method operates in three steps: object alignment, contour to multilevel image transformation and similarity evaluation. This approach is suitable for use in CBIR. The proposed technique produced encouraging results when applied on the MPEG-7 test data.

Index Terms— Content, Retrieval, Indexing, Image, Contour, Boundary, Shape, Ordinal, Correlation, Similarity, Measure.

I. INTRODUCTION

Generally shape representation can be based on its outer boundary or on the regions it contains. Characterizing the shape of an object by its boundary meets the way humans perceive objects. Since the human visual system itself concentrates on edges and ignores uniform regions [3]. This capability is hard-wired into our retinas. Connected directly to the rods and cones of the retina are two layers of the neurons that perform an operation similar to the Laplacian. This operation is called local inhibition and helps us to extract boundaries and edges [4].

Object shapes however, will have intrinsic intra-class variations. Moreover, object boundary deformation is expected in most imaging applications due to the varying imaging conditions, sensor noise, occlusion and imperfect segmentation.

Deformable models may be a promising approach for solving this problem due to their flexibility in object modeling. On the other hand, they are computationally very expensive to be used in a real time application, or even in a retrieval application where the user expects to have a response within few seconds after he puts his query. Hence these are not suitable for large databases where thousand of images are involved.

Therefore, simpler shape features have been used in several content-based indexing and retrieval (CBIR) systems, e.g. QBIC [10], MUVIS [1, 2]: high curvature points [5, 6, 7, 11], moments, morphological features (skeleton) and topological features.

This paper is introducing a novel boundary-based approach to shape similarity computation suitable for use in CBIR systems. The rest of the paper is organized as follows: Section 2 presents an overview of the proposed method, followed by a detailed description of each step. Assessments of experimental results using a subset of the MPEG-7 test data are presented in Section 3. In Section 4 conclusions are drawn.

II. THE PROPOSED METHOD

We are assuming in this paper that the shapes are already extracted from the gray level images and are stored in separate data files. The goal of this method is to compute similarity between any two shapes. The proposed method operates in three steps: alignment, boundary to multilevel image transformation and similarity evaluation.

To estimate the similarity between two objects shapes the boundaries are first aligned. The binary images containing the boundaries are then transformed into multilevel images; which are compared using the ordinal measure introduced in [9]. This ordinal measure estimates the similarity between the two shapes based on the correlation of their corresponding transform images. In the rest of this Section we give a detailed description of each one of the steps mentioned above.

A. Alignment

The alignment is performed by first detecting the major and minor axes of each shape, followed by reorientation of the shape in such a way that these axes are oriented in a standard way for all shape boundaries.

Once the major and minor axes are found, the boundary points {P1, P2, P3, and P4} that intersect with these axes are used to reorient/reposition the boundary as follows. The point that is closest to the center of mass among {P1, P2} is kept on the right. And the point among {P3, P4} closest to the major axe is kept on the top. We understand that this simple alignment process may not be enough in certain situations, but can be used to prove the validity of the proposed technique. In future work a more robust alignment algorithm may be used.

B. Boundary to multilevel image transformation

The shape is represented as a thin contour, C, in a binary image. This image is transformed into a multilevel (gray-scale) image G using a mapping function, such that the pixel values in G, $\{G_1, G_2, \ldots, G_n\}$, depend only on the position of the contour pixels:

$$G_i = \phi\,(C_k : k = 1,..., p), \text{ for } i = 1, \ldots, n,$$

where C_k is the position of the contour pixel k in the image G. Several transformations satisfy this requirement. For example any distance transformation or the transformations simulating the heat dissipation process.

As a result of this mapping the information contained in the shape boundary will be spread throughout all the pixels of the image. Computing the similarity in the transform domain will benefit of the rearrangement of the boundary information. We expect that there is no single optimal mapping; different mappings will emphasize different features of the contour. Which of these features is the most important is application and data dependent.

In this work we have implemented a mapping based on a simple geodesic metric. The metric is integer and its application is done through an iterative wave propagation process. The contour points are considered as seeds during the construction of the distance map. The distance map can be generated inside and/or outside the contour. The values can increase or decrease starting from the contour and can be limited.

Figure 2 presents an example of a distance map generated only inside the contour of a rat.

C. Similarity evaluation

The evaluation of image similarity is based on the framework for ordinal-based image correspondence introduced in [8]. Figure 3 gives a general overview of this region-based approach.

Suppose we have two images, X and Y, of equal size. In a practical setting, images are resized to a common size. Let $\{X_1, X_2, \ldots, X_n\}$ and $\{Y_1, Y_2, \ldots, Y_n\}$ be the pixels of image X and image Y, respectively. We select a number of areas $\{R_1, R_2, \ldots, R_m\}$ and extract the pixels from both images that belong to these areas. Consequently, R_j^X and R_j^Y contain the pixels from image X and Y, respectively, which belong to area R_j, with $j = 1, \ldots, m$.

The goal is to compare the two images using a region-based approach. To this end, we will be comparing R_j^X and R_j^Y, for each $j = 1, \ldots, m$. Thus, each block in one image is compared to the corresponding block in the other image in an ordinal fashion. Because our approach is an ordinal one only the ranks of the pixels are to be utilized. For every pixel X_k, we construct a so-called slice, which is defined as: $S_k^X = \{S_{k,l}^X : l = 1,..., n\}$, where:

$$S_{k,l}^X = \begin{cases} 1, \text{ if } X_k < X_l \\ 0, \quad \text{otherwise} \end{cases}.$$

As can be seen, slice S_k^X corresponds to pixel X_k and is a binary image of size equal to image X. Slices are built in a similar manner for image Y as well.

With the goal of comparing regions R_j^X and R_j^Y, we first combine the slices from image X, corresponding to all the pixels belonging to region R_j^X. The slices are combined using the operation $OP_1(.)$ into a so-called metaslice M_j^X. More formally, $M_j^X = OP_1(\{S_k^X : X_k \in R_j^X\})$, $j = 1, \ldots, m$. Similarly, we combine the slices from image Y to form M_j^Y. It should be noted that the metaslices are equal in size to the original images and could be multi-valued, depending on the operation $OP_1(.)$. Each metaslice represents the relation between the region it corresponds to and the entire image.

The next step is a comparison between all pairs of metaslices M_j^X and M_j^Y by using operation $OP_2(.)$, resulting in the metadifference D_j. That is, $D_j = OP_2(M_j^X, M_j^Y)$, $j = 1, \ldots, m$. We thus construct a set of metadifferences $D=\{D_1, D_2, \ldots, D_m\}$. The final step is to extract a scalar measure of correspondence from set D, using operation $OP_3(.)$. In other words, $\lambda = OP_3(D)$. In [8] it was shown that this structure could be used to model the well-known Kendall's τ and Spearman's ρ measures.

The image similarity measure used in this paper is an instance of the previously mentioned framework. This measure has been analyzed more extensively in [9]. Following is a short description of the operations $OP_k(.)$ adopted for this measure. Operation $OP_1(.)$ is chosen to be the component-wise summation operation; that is, metaslice M_j is the summation of all slices corresponding to the pixels in block j or in other words, $M_j = \sum_{k:X_k \in R_j} S_k$.

Next, operation $OP_2(.)$ is chosen to be the squared Euclidean distance between corresponding metaslices. That is, $D_j = \left\| M_j^X - M_j^Y \right\|_2^2$.

Finally, operation $OP_3(.)$ sums together all metadifferences to produce $\lambda = \sum_j D_j$.

One advantage of this approach over classical ordinal correlation measures is its' capability to take into account differences between images at a scale related to the chosen block size.

III. Experimental Results

The experiments were conducted on two sets of 20 images each. The two sets are from the MPEG-7 CE Shape/Motion test set B, which contains (1400 images 20 in each category).

To assess the performance of our technique the two test sets where chosen such that each contains four categories of objects.

The difference between silhouettes from any two categories of the first test set is obvious; see Figure 4. In Figure 5, this can be clearly noticed in the high levels away from the diagonal. The intra-category discrepancy is rather small. Those plateaus on the diagonal have lower levels, meaning that the shapes within a given category are well grouped.

All the images of the second test set, are side shots of animals in similar positions. Therefore, the difference (similarity) between the shapes in this set is more difficult to estimate. Especially, when a given category presents a clear difference between its members. This is the case of the deer-1 and elephant-12, see Figure 6. This difference is reflected by the larger values of the dissimilarity scores presented in Table 2.

IV. Conclusions

The proposed technique produced encouraging results when applied on the MPEG-7 test data. These results have been obtained by using some intuitively selected parameters for the generation of the mapping and the similarity evaluation. Better results are expected if the parameters are optimized.

Due to the large number of parameters involved in the generation of the distance map and in the similarity computations the behavior of the proposed technique is quite complex. Therefore, further study is required for the optimization of the results and the understanding of the method's behavior.

V. References

[1] Alaya Cheikh F., Cramariuc B., Reynaud C., Quinghong M., Dragos-Adrian B., Hnich B., Gabbouj M., Kerminen P., Mäkinen T. and Jaakkola H., "MUVIS: A System for Content-Based Indexing and Retrieval in Large Image Databases," Proceedings of the SPIE/EI'99 Conference on Storage and Retrieval for Image and Video Databases VII, Vol. 3656, pp.98-106, San Jose, California, 26-29 January 1999.

[2] Trimeche M., Alaya Cheikh F., Gabbouj M. and Cramariuc B., "Content-based Description of Images for Retrieval in Large Databases: MUVIS", X European Signal Processing Conference, Eusipco-2000, Tampere, Finland, September 5-8, 2000.

[3] Hildreth C., "The Detection of intensity changes by computer and biological vision systems", Comput. Vis. Graphics Image Proc.Vol. 22, pp. 1-27, 1983.

[4] Russ J. C., "The Image Processing Handbook", 3rd edition, CRC, Springer and IEEE Press inc., 1995.

[5] Teh C.H. and Chin R. T., "On the detection of dominant points on digital curves", IEEE Trans. PAMI, vol. 11, pp. 859-872, 1989.

[6] Koch M. W. and Kashyap R.L., " Using polygon to recognize and locate partially occluded objects", IEEE Trans. PAMI, vol. 9, pp. 483-494, 1987.

[7] Abbasi S., Mokhtarian F., and Kittler J., "Curvature Scale Space image in Shape Similarity Retrieval," Springer Journal of Multimedia Systems, 1999.

[8] Shmulevich I., Cramariuc B. and Gabbouj M., "A framework for ordinal-based image correspondence," X European Signal Processing Conference (EUSIPCO-2000), 5-8 September 2000, Tampere, Finland, pp. 1389-1392.

[9] Cramariuc B., Shmulevich I., Gabbouj M. and Makela A., "A new image similarity measure based on ordinal correlation," IEEE International Conference on Image Processing, Vancouver, BC, Canada, September 10 - 13, 2000.

[10] Niblack W. et al, "The QBIC project; querying images by content using color, texture and shape", SPIE, Vol. 1908, 1993.

[11] Quddus A., Alaya Cheikh F. and Gabbouj M., "Wavelet-Based Multi-level Object Retrieval in Contour Images," Accepted for Very Low Bit rate Video Coding (VLBV'99) workshop, October 29-30, 1999, Kyoto, Japan.

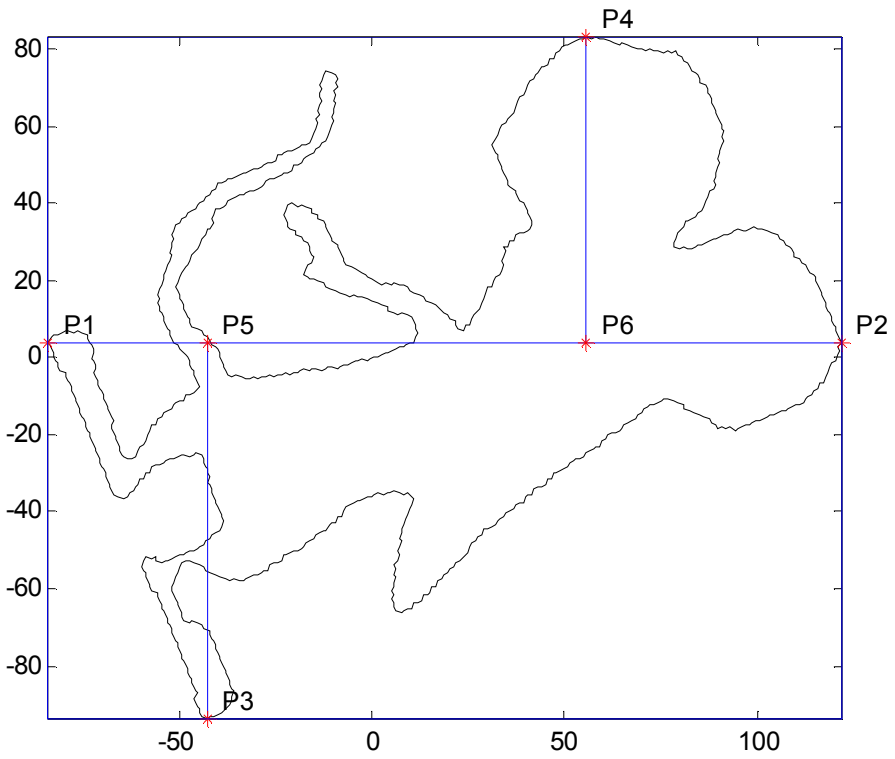


Figure 1. Boundary points used for the alignment.

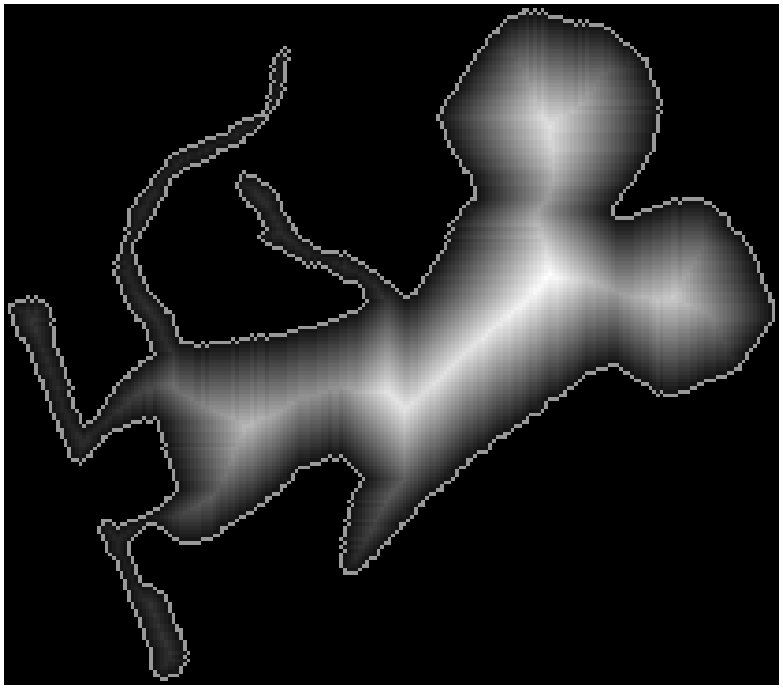


Figure 2. The object boundary overlaid on top of the distance map.

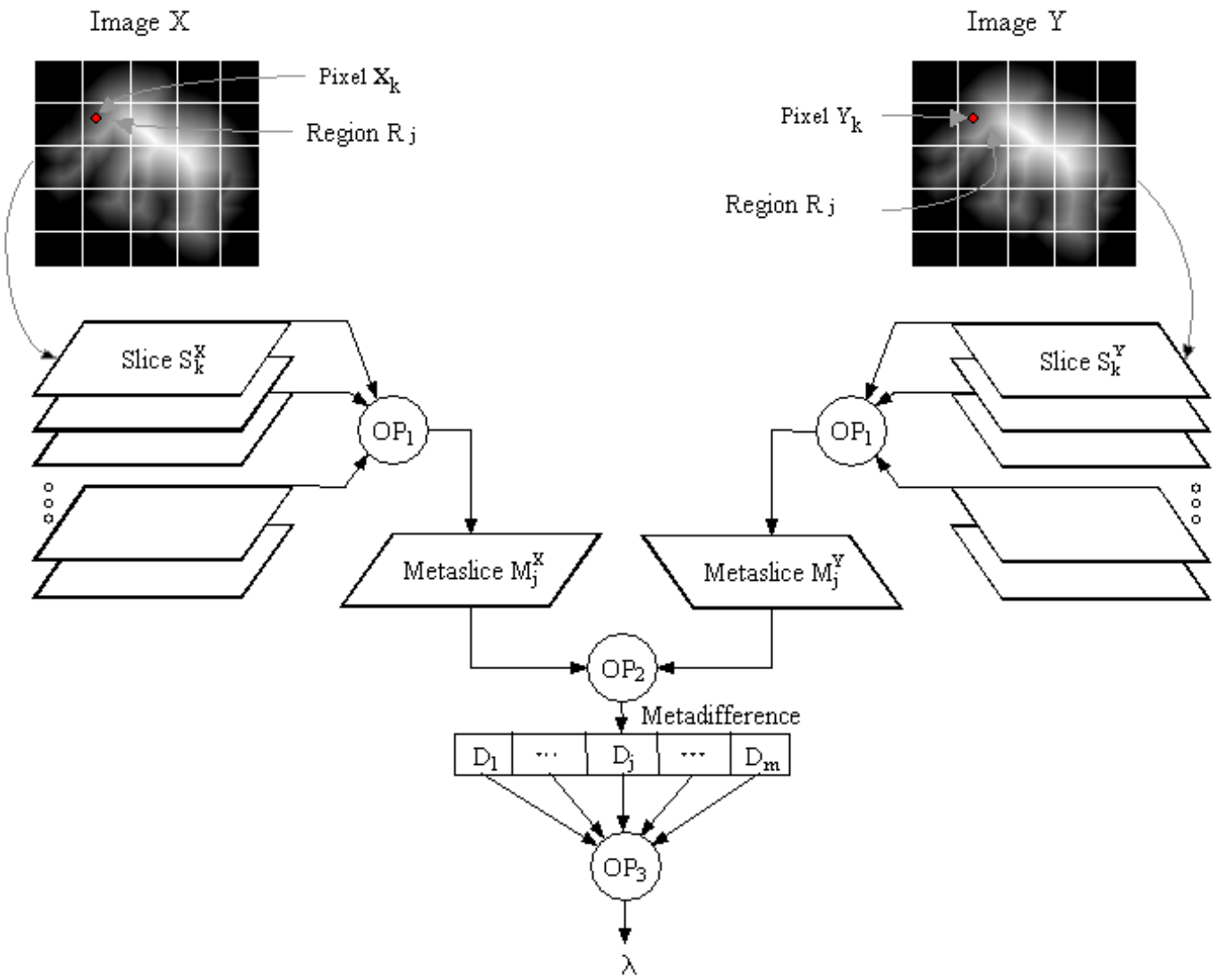


Figure 3. A general framework for ordinal-based image correspondence.

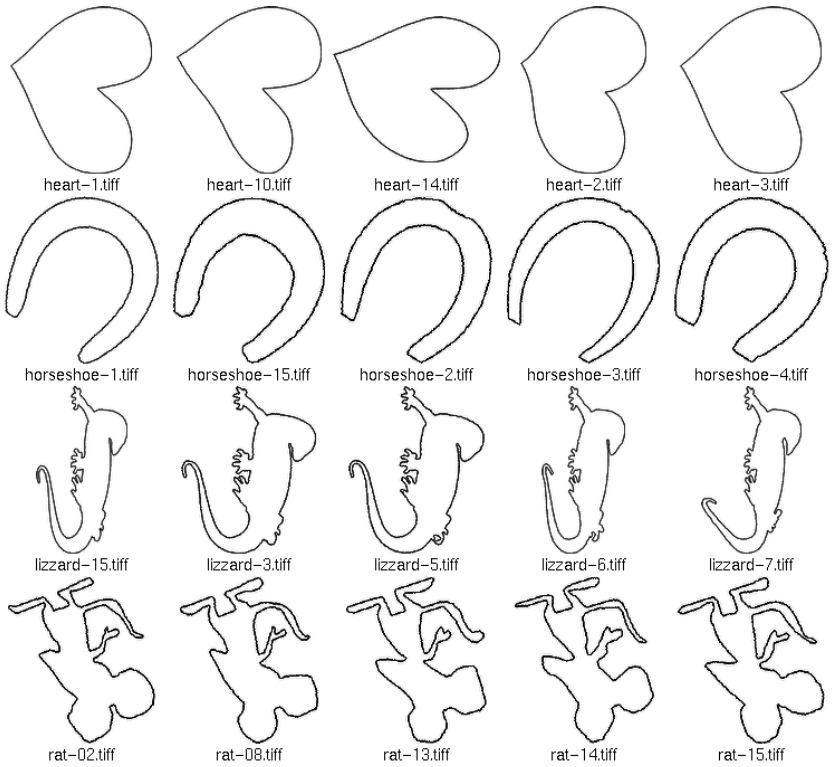


Figure4. Test set used in Experiment 1.

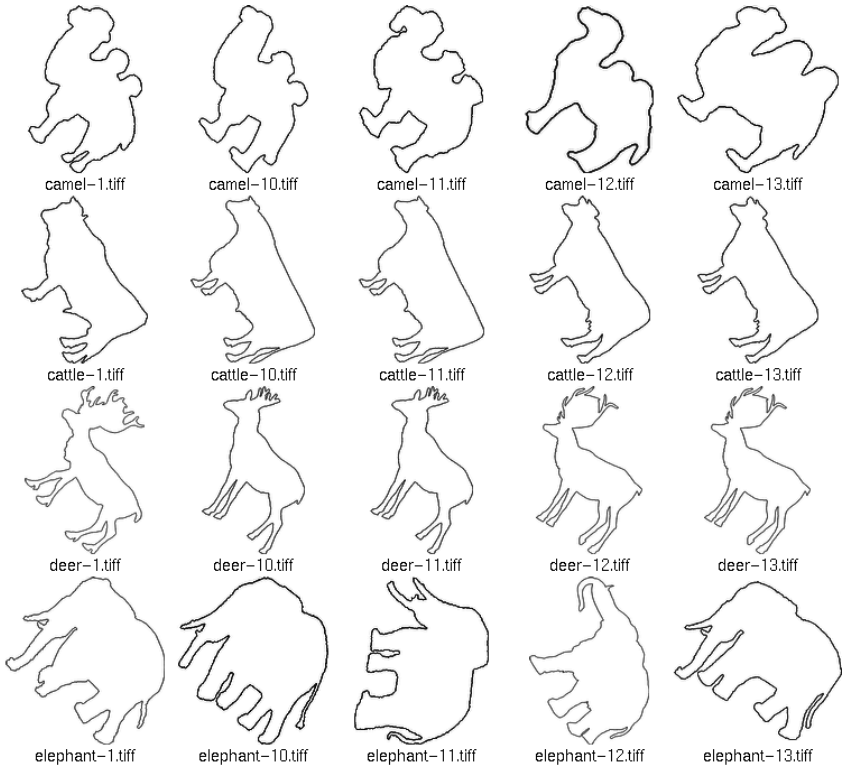


Figure 5. Test set used in Experiment 2.

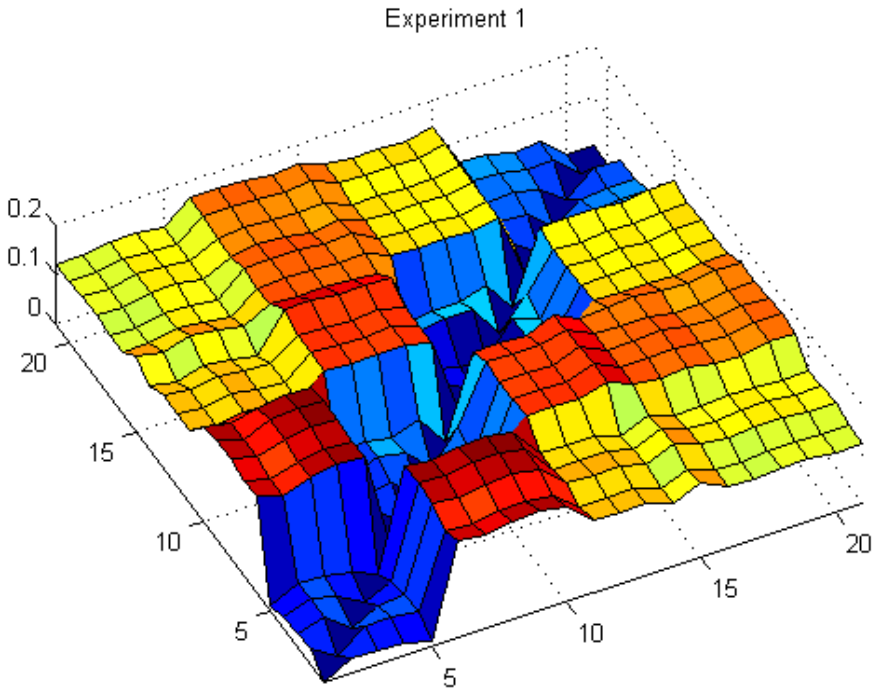


Figure 6. Surface plot of the scores obtained with the shape test set shown in **Figure 4**.

Table 1. Scores obtained for the test set number one.

Files	heart-1	heart-10	heart-14	heart-2	heart-3	horseshoe-1	horseshoe-15	horseshoe-2	horseshoe-3	horseshoe-4	lizzard-15	lizzard-3	lizzard-5	lizzard-6	lizzard-7	rat-02	rat-08	rat-13	rat-14	rat-15
heart-1	0	0.31	0.27	0.24	0.11	1.94	1.89	1.98	2.01	1.91	1.45	1.45	1.45	1.28	1.51	1.24	1.21	1.26	1.27	1.22
heart-10	0.31	0	0.42	0.42	0.35	1.78	1.73	1.83	1.83	1.79	1.38	1.39	1.39	1.19	1.4	1.23	1.19	1.23	1.24	1.22
heart-14	0.27	0.42	0	0.29	0.34	1.97	1.92	1.98	2.03	1.93	1.48	1.48	1.47	1.33	1.57	1.37	1.32	1.37	1.36	1.31
heart-2	0.24	0.42	0.29	0	0.24	2.05	1.99	2.07	2.1	2	1.4	1.4	1.4	1.26	1.5	1.31	1.29	1.35	1.34	1.3
heart-3	0.11	0.35	0.34	0.24	0	2.02	1.96	2.04	2.07	1.97	1.36	1.36	1.36	1.18	1.41	1.24	1.22	1.28	1.27	1.24
horseshoe-1	1.94	1.78	1.97	2.05	2.02	0	0.36	0.34	0.37	0.48	1.77	1.77	1.76	1.92	1.69	1.62	1.66	1.55	1.55	1.64
horseshoe-15	1.89	1.73	1.92	1.99	1.96	0.36	0	0.41	0.6	0.51	1.73	1.74	1.72	1.93	1.66	1.67	1.69	1.6	1.59	1.66
horseshoe-2	1.98	1.83	1.98	2.07	2.04	0.34	0.41	0	0.46	0.41	1.73	1.72	1.71	1.88	1.64	1.65	1.69	1.58	1.57	1.63
horseshoe-3	2.01	1.83	2.03	2.1	2.07	0.37	0.6	0.46	0	0.63	1.76	1.75	1.74	1.87	1.63	1.59	1.61	1.5	1.52	1.58
horseshoe-4	1.91	1.79	1.93	2	1.97	0.48	0.51	0.41	0.63	0	1.61	1.6	1.59	1.83	1.57	1.62	1.65	1.57	1.56	1.62
lizzard-15	1.45	1.38	1.48	1.4	1.36	1.77	1.73	1.73	1.76	1.61	0	0.2	0.19	0.56	0.37	1.39	1.36	1.39	1.34	1.42
lizzard-3	1.45	1.39	1.48	1.4	1.36	1.77	1.74	1.72	1.75	1.6	0.2	0	0.05	0.66	0.46	1.35	1.33	1.36	1.32	1.37
lizzard-5	1.45	1.39	1.47	1.4	1.36	1.76	1.72	1.71	1.74	1.59	0.19	0.05	0	0.66	0.47	1.36	1.34	1.36	1.31	1.38
lizzard-6	1.28	1.19	1.33	1.26	1.18	1.92	1.93	1.88	1.87	1.83	0.56	0.66	0.66	0	0.62	1.33	1.3	1.34	1.31	1.36
lizzard-7	1.51	1.4	1.57	1.5	1.41	1.69	1.66	1.64	1.63	1.57	0.37	0.46	0.47	0.62	0	1.42	1.41	1.42	1.38	1.47
rat-02	1.24	1.23	1.37	1.31	1.24	1.62	1.67	1.65	1.59	1.62	1.39	1.35	1.36	1.33	1.42	0	0.37	0.45	0.46	0.56
rat-08	1.21	1.19	1.32	1.29	1.22	1.66	1.69	1.69	1.61	1.65	1.36	1.33	1.34	1.3	1.41	0.37	0	0.44	0.51	0.54
rat-13	1.26	1.23	1.37	1.35	1.28	1.55	1.6	1.58	1.5	1.57	1.39	1.36	1.36	1.34	1.42	0.45	0.44	0	0.38	0.41
rat-14	1.27	1.24	1.36	1.34	1.27	1.55	1.59	1.57	1.52	1.56	1.34	1.32	1.31	1.31	1.38	0.46	0.51	0.38	0	0.45
rat-15	1.22	1.22	1.31	1.3	1.24	1.64	1.66	1.63	1.58	1.62	1.42	1.37	1.38	1.36	1.47	0.56	0.54	0.41	0.45	0

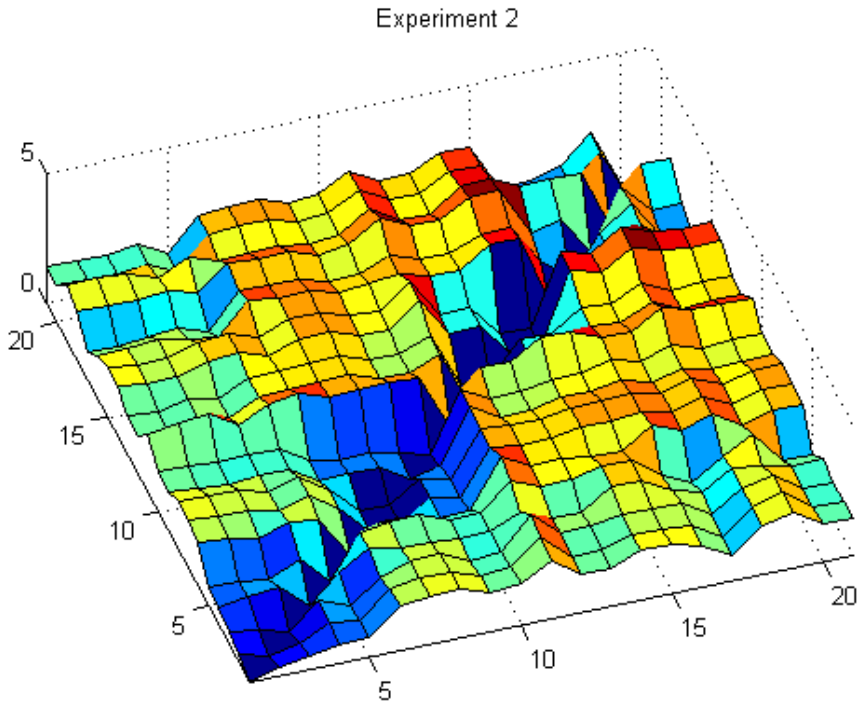


Figure 7. Surface plot of the scores obtained with the shape test set shown in **Figure 5**.

Table 2. Scores obtained for the test set number two.

Files	camel-1	camel-10	camel-11	camel-12	camel-13	cattle-1	cattle-10	cattle-11	cattle-12	cattle-13	deer-1	deer-10	deer-11	deer-12	deer-13	elephant-1	elephant-10	elephant-11	elephant-12	elephant-13
camel-1	0	0.36	0.51	0.68	0.74	1.75	1.87	1.88	1.50	1.57	2.23	1.47	1.48	1.92	1.92	1.60	1.01	1.87	2.14	1.45
camel-10	0.36	0	0.32	0.83	0.68	1.61	1.79	1.80	1.41	1.49	2.46	1.42	1.44	1.75	1.75	1.67	1.00	1.88	2.18	1.36
camel-11	0.51	0.32	0	0.97	0.53	1.36	1.76	1.76	1.28	1.40	2.55	1.61	1.63	1.81	1.81	1.75	1.13	1.82	2.12	1.34
camel-12	0.68	0.83	0.97	0	1.09	1.77	1.49	1.50	1.36	1.44	1.89	1.42	1.42	2.14	2.14	1.56	1.25	1.92	1.71	1.51
camel-13	0.74	0.68	0.53	1.09	0	1.61	1.79	1.80	1.36	1.43	2.42	1.95	1.96	2.23	2.23	1.37	0.83	1.62	2.25	0.97
cattle-1	1.75	1.61	1.36	1.77	1.61	0	1.05	1.05	0.73	0.78	2.62	2.01	2.03	1.86	1.86	2.57	2.15	2.53	1.89	2.17
cattle-10	1.87	1.79	1.76	1.49	1.79	1.05	0	0.01	0.74	0.54	2.23	2.07	2.08	2.29	2.29	2.38	2.13	2.45	1.94	2.24
cattle-11	1.88	1.80	1.76	1.50	1.80	1.05	0.01	0	0.74	0.54	2.23	2.08	2.09	2.29	2.29	2.39	2.14	2.45	1.95	2.25
cattle-12	1.50	1.41	1.28	1.36	1.36	0.73	0.74	0.74	0	0.32	2.33	1.70	1.71	1.94	1.94	2.10	1.81	2.09	1.81	1.88
cattle-13	1.57	1.49	1.40	1.44	1.43	0.78	0.54	0.54	0.32	0	2.24	1.68	1.69	1.90	1.90	2.19	1.87	2.30	1.91	2.00
deer-1	2.23	2.46	2.55	1.89	2.42	2.62	2.23	2.23	2.33	2.24	0	2.14	2.14	2.79	2.79	2.33	2.30	3.01	2.68	2.61
deer-10	1.47	1.42	1.61	1.42	1.95	2.01	2.07	2.08	1.70	1.68	2.14	0	0.03	1.22	1.22	2.03	1.91	2.49	1.96	2.05
deer-11	1.48	1.44	1.63	1.42	1.96	2.03	2.08	2.09	1.71	1.69	2.14	0.03	0	1.27	1.27	2.01	1.91	2.47	1.95	2.04
deer-12	1.92	1.75	1.81	2.14	2.23	1.86	2.29	2.29	1.94	1.90	2.79	1.22	1.27	0	0	2.73	2.40	3.12	2.80	2.62
deer-13	1.92	1.75	1.81	2.14	2.23	1.86	2.29	2.29	1.94	1.90	2.79	1.22	1.27	0	0	2.73	2.40	3.12	2.80	2.62
elephant-1	1.60	1.67	1.75	1.56	1.37	2.57	2.38	2.39	2.10	2.19	2.33	2.03	2.01	2.73	2.73	0	0.91	1.24	2.27	0.74
elephant-10	1.01	1.00	1.13	1.25	0.83	2.15	2.13	2.14	1.81	1.87	2.30	1.91	1.91	2.40	2.40	0.91	0	1.46	2.36	0.84
elephant-11	1.87	1.88	1.82	1.92	1.62	2.53	2.45	2.45	2.09	2.30	3.01	2.49	2.47	3.12	3.12	1.24	1.46	0	2.29	1.13
elephant-12	2.14	2.18	2.12	1.71	2.25	1.89	1.94	1.95	1.81	1.91	2.68	1.96	1.95	2.80	2.80	2.27	2.36	2.29	0	2.15
elephant-13	1.45	1.36	1.34	1.51	0.97	2.17	2.24	2.25	1.88	2.00	2.61	2.05	2.04	2.62	2.62	0.74	0.84	1.13	2.15	0

FAST USER-ADAPATIVE WEIGHTING OF MPEG7 DESCRIPTORS FOR A VISUAL E-COMMERCE INTERFACE

Ivo Keller, Thomas Ellerbrock, Thomas Meiers, Thomas Sikora,
Heinrich-Hertz-Institute for Communication, Einsteinufer 37, D-14195 Berlin, GERMANY
Tel.: +49 30 31002-387, Fax.: +49 30 31002-212
email: keller@hhi.de, ellerbrock@hhi.de, meiers@hhi.de, sikora@hhi.de

ABSTRACT

E-Commerce systems must offer customers a quick and easy access to products. In this paper we propose a visualization system allowing a convenient overview of large sets of objects. In the beginning, the product's visual features are described in terms of very high-dimensional MPEG-7 descriptors and are reduced to only three dimensions for visual presentation afterwards. The dimension reduction is realized by an appropriate weighting of the high-dimensional descriptor components corresponding to a modification of the covariance-matrix used for PCA. The user can adapt the visualization procedure according to his personal preferences in a continuous and intuitive manner in real time. This is regarded to be of great importance for the acceptance of E-Commerce systems. The technique introduced is a general approach, which can be combined with other relevance feedback methods.

1 INTRODUCTION

E-Commerce is regarded to be one of the most promising factors of industrial growth and companies all over the world invest more and more into this field.
To be successful, e-commerce systems must present products in an appealing manner, allow easy and intuitive navigation through the range of goods and highlight their characteristics and merits.
This is not an easy task, especially if there is a large diversity of products and if goods cannot sufficiently be described by text or numbers (e.g. wallpaper or knobs).

Every customer in a store uses a high-performance pattern recognition system to find the goods he is looking for: his own visual system. When searching for goods the customer is guided by his idea of how the desired product may look or even by a sample of the product. Visual similarity and visual order are major factors in searching. Therefore, it is little surprising that knobs in the haberdashery of a department store, for example, are usually sorted according to their shape, color and size. The efficiency of the human visual system allows the customer to quickly scan and match a vast amount of objects based on visual criteria and select the most promising ones for a closer inspection.

The MPEG-7 descriptors for still image- and video description are well adapted to visual feature extraction and are also suited for fast indexing. Each descriptor covers a part of a high dimensional feature space, which may achieve up to 300 or even more dimensions.
For visualization the feature space has to be reduced to 2 or 3 dimensions. We use Principal Component Analysis (PCA) here for dimension reduction. The computation of the covariance matrix is a time consuming procedure, but it has to be done only once after the database has been created. So its computation does not effect the e-commerce system at run time.
The PCA yields an uncorrelated feature vector representation with minimum difference between the original feature vector and the feature vector which is reconstructed from the dimensionally reduced one [3]. This is achieved keeping only those components of the transformed feature vector having the highest variance.
However, components being salient in a pure statistical sense might not be relevant to the user. Rather the user may be interested in some particular aspects with minor statistical importance. Therefore, a user-specific weighting of feature components is needed to take into account personal preferences.
In contrast to different interactive systems our e-commerce interface enables fast visualization, easy use and intuitive presentation. MARS [11] for example, asks the user to rate the search results. In [10] a system is introduced which applies learning algorithms for data, task and context recognition. Although appealing, these approaches are somewhat circumstantial and not suited for easy and quick application.

User feedback can be integrated by weighting each descriptor (feature vector, resp.) component by some factor followed by a new computation of the entire PCA. With interactive systems, the user permanently reacts to system output and permanently changes user-specific parameters. In chapter 2 we show how real time user feedback can be processed without recomputation of the entire covariance matrix. In Chapter 3, we demonstrate the usefulness of PCA for the visualization of selected MPEG-7 descriptors. These descriptors can be viewed to be the search intention of a potential costumer. For demonstration three descriptors are selected: *BoundingBox*, *ContourBasedShape* and *RegionBasedShape*. They are applied to a database containing a variety of shape objects

2 DIMENSION REDUCTION

As described above, the first step is dimension reduction. Non-linear techniques like Neural Networks (especially Multilayer Perceptrons and Self Organizing Maps [4], [9] [1]) provide a better adaption to heavily folded clusters in the feature space than PCA. However, the integration and adjustment of user-specific weights is hard to realize. Therefore, PCA is used here.

PCA is also known as "Karhunen-Loeve Transform" (KLT). It rotates and shifts the coordinate system in such way that features are made uncorrelated. After this, feature components with small variance are neglected. This corresponds to a projection into a low dimensional subspace capturing most of the variation within the data set [7].

The PCA (for details see [2,3,7]) takes a set of real n-dimensional (feature) vectors as input. Let $x^{(1)},\dots,x^{(N)}$ denote the feature vectors, which result from the original ones after substraction of their mean (this way the PCA shifts its basis transformation to the center of mass of the given set of vectors). After this, the covariance matrix C is the expectation of the matrix $X_{ij} := x_i \cdot x_j$, i.e.

$$C = \mathrm{E}(X) \qquad (1)$$

Lets $v^{(1)},\dots,v^{(n)}$ be the normalized eigenvectors of C, arranged in decreasing order of their eigenvalues $\lambda^{(1)},\dots,\lambda^{(n)}$, and define the projection matrix V to be an $n \times k$ matrix having the eigenvectors $v^{(1)},\dots,v^{(k)}$ as columns. The vectors $v^{(1)},\dots,v^{(n)}$ are an orthonormal basis of the n-dimensional euclidean feature space. For each original feature vector x a k-dimensional feature vector y is computed by application of the transposed matrix of V to x, i.e.

$$y = V^{\mathrm{T}}x \qquad (2)$$

$$\text{with } k \ll n,\ k \in \{2,3\} \text{ here.} \qquad (3)$$

The eigenvalues $\lambda^{(1)},\dots,\lambda^{(k)}$ are the variances of the new feature vectors y. Note, that y is a vector of a reduced number of dimensions corresponding to the largest variances. The vector y can be regarded as a mixture of components of the original feature vector x.

2.1 Integration of user-intention

There are two possibilities to integrate user preference weights into the PCA. The first one virtually changes the variance of the data by rescaling the axes. After this is done the covariance matrix C must be computed again. Alternatively, only the columns and rows concerned are adjusted. Suppose, the m-th component of all zero mean feature vectors $x^{(1)},\dots,x^{(N)}$ shall be emphasized by a factor α (this corresponds to an increment of the variance of dimension m.)

$$x'^{(n)}(m) := \alpha \cdot x^{(n)}(m),\ \forall n \in \{1,\dots,N\}, \qquad (4)$$

However, this corresponds to a modification C' of the covariance matrix C:

$$C'(i,j) = \begin{cases} C(i,j), & i,j \neq m \\ \alpha \cdot C(i,j), & i = m \vee j = m \\ \alpha^2 \cdot C(i,j), & i = j = m \end{cases} \qquad (5)$$

Due to eq. (5) the recomputation of the covariance matrix can be avoided. Instead only some columns and rows are multiplied by weights. However, this does not prevent from a new computation of eigenvectors and eigenvalues.

It is obvious from the formulas above, that by the introduced approach, user preferences control the principal components chosen within the PCA. This enables the user to visualize those subspaces which appear relevant to him.

3 EXAMPLES OF USER-SPECIFIC SEARCH

The efficiency of the user-interactive system is demonstrated for a database storing numerous objects of different kind. Each object is assigned a 3-dimensional PCA-based feature vector y. This vector is used to place the objects within a virtual 3D-space. The e-commerce user-interface depicts all the objects in this virtual 3D-space. To enhance the 3D impression the ensemble of objects is rotated within the virtual space resulting in a strong 3D illusion. This way many more objects can be viewed and selected than in two dimensions. By means of a few sliders the user is able to control search items and subspace axis in the high-dimensional feature space corresponding to "orientation", "contour" and "shape". These sliders adjust user preference weights and correspond to the rescaling of the original feature space spanned by the MPEG-7 descriptors.

3.1 The shape database

The database used contains about 3500 examples of shapes, mostly varied in perspective. It has been provided by the MPEG-7 community and is commonly used for the evaluation of descriptors. Shape visualization has also been proposed for the recognition of trademarks [5]. In principal, any feature extraction algorithm can be integrated into this visualization tool provided the features can be considered elements of some vector space.

We have investigated three shape-based MPEG-7 descriptors: *BoundingBox*, *ContourBasedShape*, and Region-

BasedShape ([8], [6], and [12]). After some transformation, the descriptors return features being elements of the 60-dimensional euclidean vector space. These 60-dimensional vectors are projected into a three dimensional subspace as explained in the former section.

3.1.1 MPEG-7 BoundingBox

Fig. 1**:** Visualization with *BoundingBox* being the dominant feature

The *BoundingBox* descriptor extracts the smallest enclosing rectangle and its orientation. It is simple but robust and might - for example - separate lying persons from standing ones.
At the first glance the user might concentrate himself to the orientation of objects and to the ratio of the object's width to height. Then he will pay most attention to visual aspects being captured by the *BoundingBox* descriptor. Figure 1 displays such a szenario. Note that objects with approximately a quadratic bounding box are placed on the right side whereas objects embraced by an elongated rectangle are on the left.
In addition the tilt angle is varied from the bottom to the top. Although 200 objects are depicted, a good overview is given in the 3D visualization.

3.1.2 MPEG-7 CountourBasedShape

This descriptor computes the maxima of the contour' s curvature. Firstly, equidistant sample points are placed on the contour to make the descriptor independent of the object size. At these points the curvature is considered and its extrema are tracked through a continuing low pass

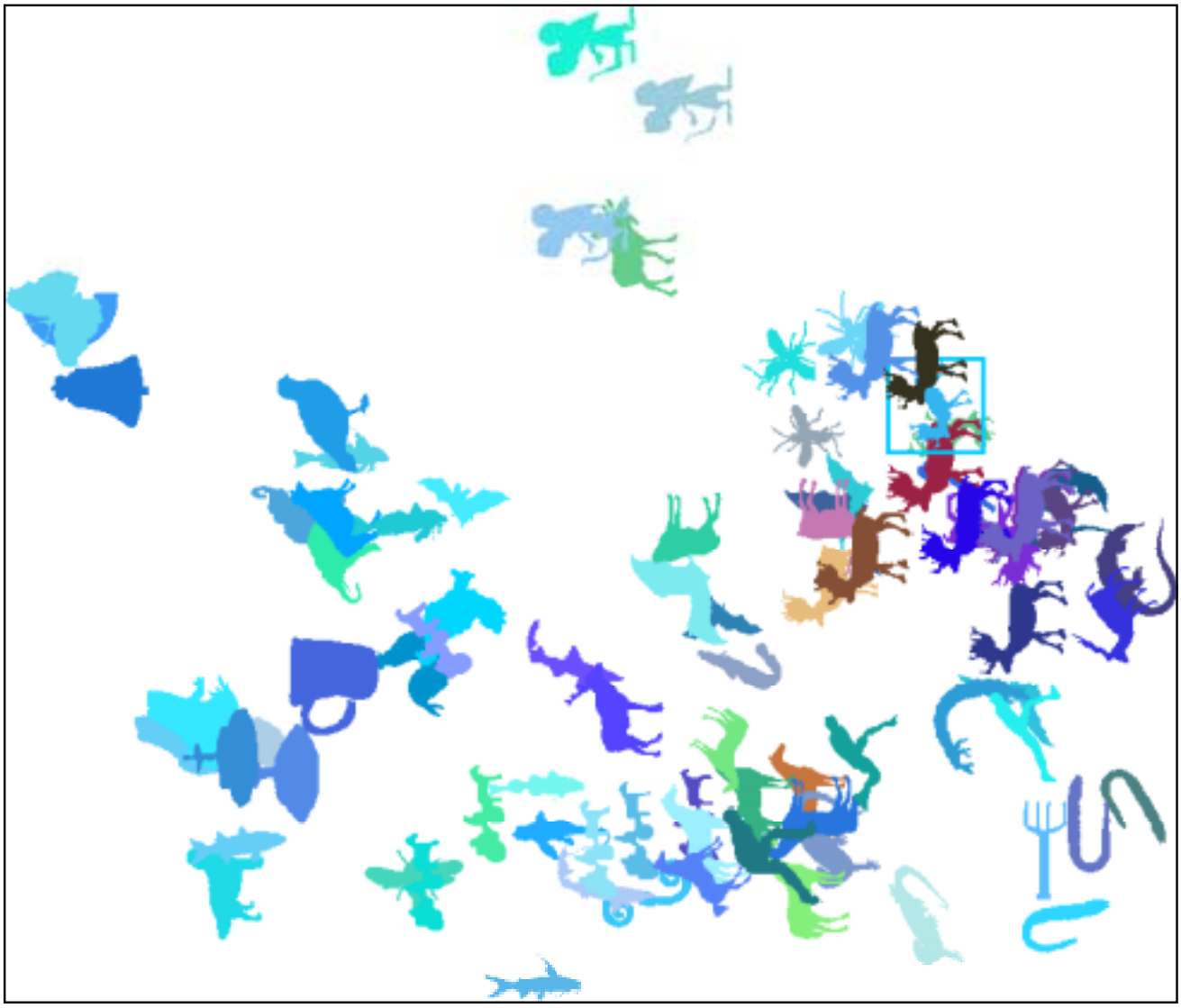


Fig. 2: Visualization with user-preference *ContourBasedShape* being the dominant feature

filtering process. When the filtering proceeds flat extrema disappear whereas distinct ones persist.

The locations of the curvature's extrema and the number of filtering steps are taken as feature vector. In addition, the data are adjusted to the highest maximum to gain some rotational invariance. The method is robust with respect to noise and it reflects the locations of salient curvature points irrespective of shape complexity. Figure 2 depicts a subset of the database. The objects concerned originate from a vicinity of shapes with quadratic BoundingBox and are arranged with respect to the fineness of theirs contours.

3.1.3 MPEG-7 RegionBasedShape

The center of mass of each object is calculated and a certain kind of radial symmetric polynomials, called Zernike Polynomials, are fitted to the centered object. This method yields invariance with respect to rotation, size, and translation.

Figure 3 shows the alignment of objects according to their mass distribution. It is remarkable that Figure 3 depicts a 3d-visualization on a 2d-surface. In the upper right part compact shapes are located whereas in the lower part objects with rather a spreading mass are situated. In contrast, even more straddle shapes are in the left half, e.g. stars, cruises, etc. Stars are subdivided into ones with straight and titled rays.

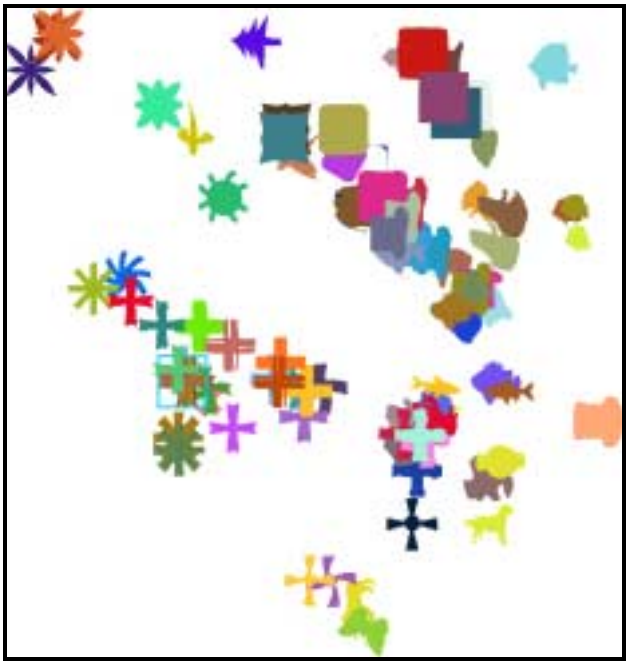


Fig. 3**:** Visualization with user-preference *RegionBased-Shape* being the dominant feature

3.1.4 Transition from BoundingBox to CountourShape

Figures 4a, 4b, 4c, 4d demonstrate a continuous transition from a *BoundingBox* dominated visualization to a visualization governed by the *ContourBasedShape* descriptor. This reflects a gradual shift of the user's preferences starting at shape proportion and shape orientation and arriving at a point of view guided by contour complexity in the end.

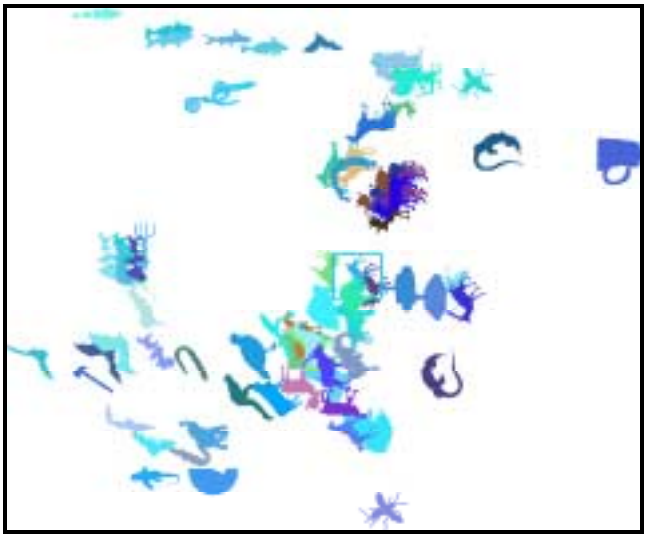


Fig. 4a: Continuous transition of feature weightings:
a) 100% *BoundingBox* and 0% *ContourBasedShape*

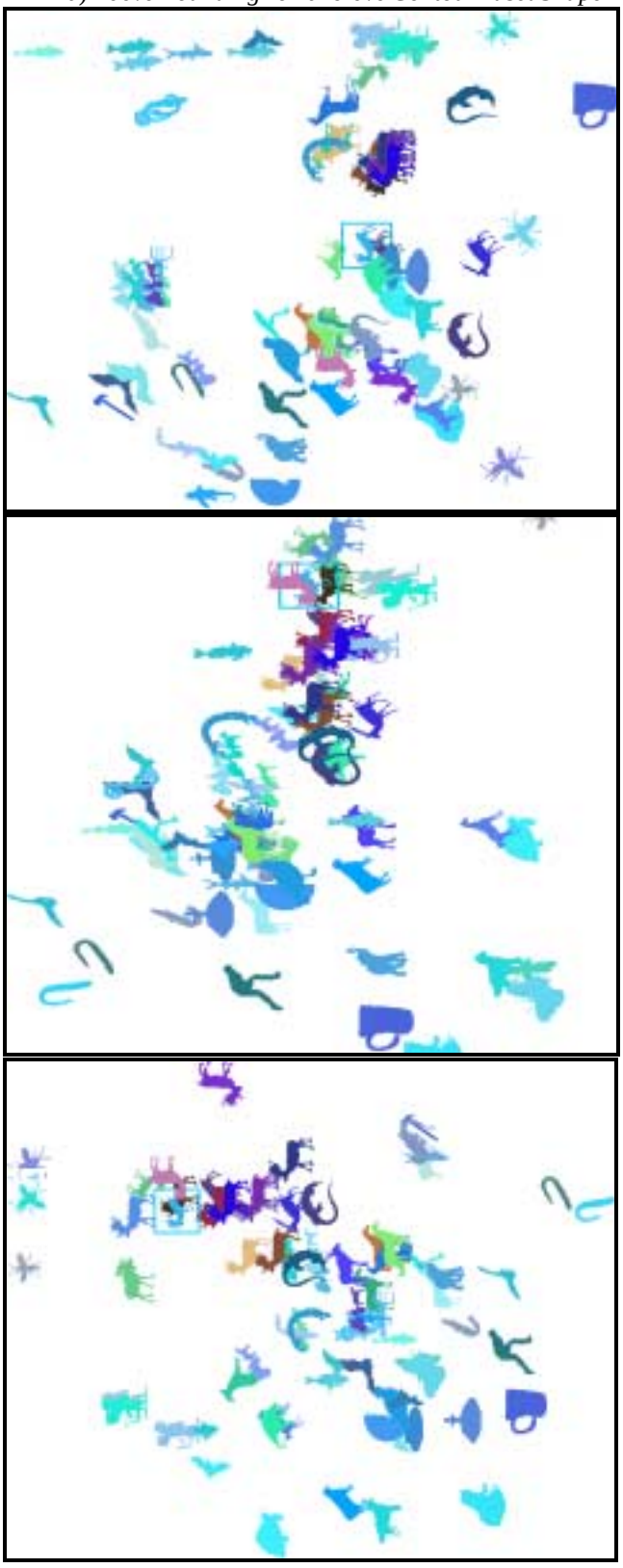


Fig. 4b,4c,4d: Continuous transition of feature weightings:
b) (top): 66% *BoundingBox* and 33% *ContourBasedShape*

c) (in the middle): 33% *BoundingBox* and 66% *ContourBasedShape*
d) (bottom): 0% *BoundingBox* and 100% *ContourBasedShape*

4 CONCLUSION AND FUTUR WORK

We have shown that PCA can be applied for the visulization of high-dimensional MPEG-7 descriptors. It yields an elegant method for effective user-interaction for e-commerce systems. Though the PCA is a linear technique, its performance is powerful and sufficient for the compression of high dimensional feature spaces to only 3 dimensions. Usually, the introduction of a user-specific weighting will require a time consuming recomputation of the PCA' s covariance matrix. However, we found quite an easy way to circumvent this problem.
Hence, our e-commerce surface allows a continuous variation of the visualized subspace in accordance with the search preferences of the user.
Future work will combine different techniques (relevance feedback, subspace visualization, 3D-viewers) to make the layman still more unaffected by the technical machínery of the software. This is regarded one of the most important factors for the acceptance of e-commerce platforms.

REFERENCES

[1] T.M. Ellerbrock, Multilayer Neural Networks: Learnability, Network Generation, and Network Simplification, Ph. D. thesis, Universität Bielefeld, Germany, 1999

[2] T.M. Ellerbrock, T. Meiers, T. Sikora, "Comments on 2nd Order Eigenface Method of M5750, M6001", MPEG-7 Video ISO/IEC JTC1/ SC29/ WG11, M6364, July 2000

[3] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, San Diego, 1990, ISBN 0-12-269851-7

[4] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the theory of Neural Computation, Addison-Wesley Publishing Company, Redwood City, USA, ISBN 0-201-50395-6

[5] Jeannin, S., Bober, M., "Description of core experiments for MPEG-7 motion/shape, Doc. ISO/IEC JTC1/ SC29/ WG11, N2690, 47th MPEG Meeting, Seoul, March 1999

[6] Jeannin, S. "MPEG-7 Visual part of eXperimentation Model Version 4.0", MPEG-7 Video ISO/IEC JTC1/ SC29/ WG11, N3068, Dec. 1999

[7] I.T. Jolliffe, Principal Component Analysis, Springer, Berlin, 1986

[8] Kim,W.Y., Kim, Y.S., "A region-based shape descriptor using Zernike moments", IEEE Signal Proc., "Image Communication", Special Issue on MPEG-7 technology, Elsevier, Vol. 16, Nos. 1-2, Sept. 2000, ISSN 0923-5965, pp. 95-102

[9] S.Y. Kung, Digital Neural Networks, Prentice Hall, Englewood Cliffs, New Jersey, 1993, ISBN 0-13-612326-0

[10] Minka, T.P., Picard, R.W., Interactive learning using a society of models, MIT Media Laboratory Perceptual Computing Section Technical Report No. 349, Submitted to Special Issue of Pattern Recognition on Image Database: Classification and Retrieval, picard@media.mit.edu, 2000

[11] Ortega, M., Rui, Y., Chakrabarti, K., Mehrotra, S., Huang, T.S., "Supporting Similarity Queries in MARS", ACM Multimedia 97, Seattle, USA, Nov. 8-14 1997

[12] Yamada, A., Pickering, M., Jeannin, S., Cieplinski, L., Ohm, J.R., Kim, M., "MPEG-7 Visual part of Experimentation Model Version", MPEG-7 Video ISO/IEC JTC1/ SC29/ WG11, M6808, Jan. 2001

AUDIO CLASSIFICATION IN SPEECH AND MUSIC: A COMPARISON OF DIFFERENT APPROACHES

A. Bugatti, A. Flammini, R. Leonardi, D. Marioli, P. Migliorati, C. Pasin

Dept. of Electronics for Automation, University of Brescia,
Via Branze 38, I-25123 Brescia – ITALY
Tel: +39 030 3715433; fax: +39 030 380014
e-mail: leon@ing.unibs.it

ABSTRACT

This paper presents a comparison between different techniques for audio classification into homogeneous segments of speech and music. The first method is based on Zero Crossing Rate and Bayesian Classification (ZB), and it is very simple from a computational point of view. The second approach uses a Multi Layer Perceptron network (MLP) and requires therefore more computations. The performance of the proposed algorithms has been evaluated in terms of misclassification errors and precision in music-speech change detection. Both the proposed algorithms give good results, even if the MLP shows the best performance.

1. INTRODUCTION

Effective navigation through multimedia documents is necessary to enable widespread use and access to richer and novel information sources. Design of efficient indexing techniques to retrieve relevant information is another important requirement. Allowing for possible automatic procedures to semantically index audio-video material represents a very important challenge. Such methods should be designed to create indices of the audio-visual material, which characterize the temporal structure of a multimedia document from a semantic point of view.

The International Standard Organization (ISO) started in October 1996 a standardization process for the description of the content of multimedia documents, namely MPEG-7: the "Multimedia Content Description Interface" [1]. However the standard specifications do not indicate methods for the automatic selection of indices.

A possible mean is to identify series of consecutive segments, which exhibit a certain coherence, according to some property of the audio-visual material. By organizing the degree of coherence, according to more abstract criteria, it is possible to construct a hierarchical representation of information, so as to create a Table of Content description of the document. Such description appears quite adequate for the sake of navigation through the multimedia document, thanks to the multi-layered summary that it provides [2,3].

Traditionally, the most common approach to create an index of an audiovisual document has been based on the automatic detection of the changes of camera records and the types of involved editing effects. This kind of approach has generally demonstrated satisfactory performance and lead to a good low-level temporal characterization of the visual content. However the reached semantic level remains poor since the description is very fragmented considering the high number of shot transitions occurring in typical audio-visual programs.

Alternatively, there have been recent research efforts to base the analysis of audiovisual documents by a joint audio and video processing so as to provide for a higher level organization of information. In [2,4] these two sources of information have been jointly considered for the identification of simple scenes that compose an audiovisual program. The video analysis associated to cross-modal procedures can be very computationally intensive (by relying, for example, on identifying correlation between non-consecutive shots).

We believe that audio information carries out by itself a rich level of semantic significance. The focus of this contribution is to compare simple classification schemes for audio segments. Accordingly, we propose and compare the performance of two different approaches for audio classification into homogeneous segments of speech and music. The first approach, based mainly on Zero Crossing Rate (ZCR) and Bayesian Classification, is very simple from a computational complexity point of view. The second approach, based on Neural Networks (specifically a Multi Layer Perceptron, MLP), allows better performance at the expense of an increased computational complexity.

The paper is organized as follows. Section 2 is devoted to a brief description of the state of the art solutions for audio classification into speech and music. The proposed algorithms are described in Sections 3 and 4, whereas in Section 5 we report the experimental results. Concluding remarks are given in Section 6.

2. STATE OF THE ART SOLUTIONS

In this section we focus the attention on the problem of speech from music separation.

J. Saunders [5] proposed a method mainly based on the statistical parameters of the Zero Crossing Rates (ZCR, plus a measure of the short time energy contour. Then, using a multivariate Gaussian classifier, he obtained a good percentage of class discrimination (about 90%). This approach is successful for discriminating speech from music on a broadcast FM radio and it allows achieving the goal for the low computational complexity and for the relative homogeneity of this type of audio signal.

E. Scheirer and M. Slaney [6] developed another approach to the same problem, which exploits different features still achieving similar results. Even in this case the algorithm achieves real-time performance and uses time domain features (short-term energy, zero crossing rate) and frequency domain features (4 Hz Modulation energy, Spectral Rolloff point, centroid and flux, ...), extracting also their variance in one second segments. In this case, they use some methods for the classification (Gaussian mixture model, k-nearest neighbor, k-d tree) and they obtain similar results.

J. Foote [7] adopted a technique purely data-driven and he did not extract subjectively "meaningful" acoustic parameters. In his work, the audio signal is first parameterized into Mel-scaled cepstral coefficients plus an energy term, obtaining a 13-dimensional feature vector (12 MFCC plus energy) at a 100 Hz frame rate. Then using a tree-based quantization the audio is classified into speech, music and no-vocal sounds.

C. Saraceno [8] and T. Zhang et al. [9] proposed more sophisticated approaches to achieve a finest decomposition of the audio stream. In both works the audio signal is decomposed at least in four classes: silence, music, speech and environmental sounds.

In the first work, at the first stage, a silence detector is used, which divides the silence frames from the others with a measure of the short time energy. It considers also their temporal evolution by dynamic updating of the statistical parameters and by means of a finite state machine, to avoid misclassification errors. Hence the three remaining classes are divided using autocorrelation measures, local as well as contextual and the ZCR, obtaining good results, where misclassifications occur mainly at the boundary between segments belonging to different classes.

In [9] the classification is performed at two levels: a coarse level and a fine level. For the first level, it is used a morphological and statistical analysis of energy function, average zero crossing rate and the fundamental frequency. Then a ruled-based heuristic procedure is built to classify audio signals based on these features. At the second level, further classification is performed for each type of sounds. Because this finest classification is inherently semantic, for each class could be used a different approach. In this work the focus is primarily on the environmental sounds which are discriminated using periodic and harmonic characteristics. The results for the coarse level show an accuracy rate of more than 90% and misclassification usually occurs in hybrid sounds, which contains more than one basic type of audio.

Z. Liu et al. [10] use another kind of approach, because their aim is to analyze the audio signal for a scene classification of TV programs. The features selected for this task are both time and frequency domain and they are meaningful for the scene separation and classification. These features are: no silence ratio, volume standard deviation, volume dynamic range, frequency component at 4 Hz, pitch standard deviation, voice of music ratio, noise or unvoiced ratio, frequency centroid, bandwidth and energy in 4 sub-bands of the signal. Feedforward neural networks are used successfully as pattern classifiers in this work. Better performances are achieved using a one-class-in-one-network (OCON) neural network rather than an all-class-in-one-network (ACON) neural network. The recognized classes are advertisement, basketball, football, news, weather forecasts and the results show the usefulness of using audio features for the purpose of scene classifications.

An alternative approach in audio data partitioning consists in a supervised partitioning. The supervision concerns the ability to train the models of the various clusters considered in the partitioning. In literature, the Gaussian mixture models (GMMs) [11] are frequently used to train the models of the chosen clusters. From a reference segmented and labeled database, the GMMs are trained on acoustic data for modeling characterized clusters (e.g., speech, music and background).

The great variability of noises (e.g., rumbling, explosion, creaking) and of music (e.g., classic, pop) observed on the audio-video databases (e.g., broadcast news, movie films) makes difficult an a priori training of the models of the various clusters characterizing these sounds. The main problem to train the models is the segmentation/labeling of large audio databases allowing a statistical training. So long as the automatic partitioning isn't perfect, the labeling of databases is time consuming of human experts. To avoid this cost and to cover the processing of any audio document, the characterization must be generic and an adaptation of the techniques of data partitioning on the audio signals is required to minimize the training of the various clusters of sounds.

3. ZCR WITH BAYESIAN CLASSIFIER

As previously mentioned, several researches assume an audio model composed of four classes: silence, music,

speech and noise.

In this work we focus the attention on the specific problem of audio classification in music and speech, assuming that the silence segments have already been identified using, e.g., the method proposed in [4].

For this purpose we use a speech characteristic to discriminate it from the music; the speech shows a very regular structure where the music doesn't show it. Indeed, the speech is composed by a succession of vowels and consonants: while the vowels are high energy events with the most of the spectral energy contained at low frequencies, the consonant are noise-like, with the spectral energy distributed more towards the higher frequencies.

Saunders [5] used the Zero Crossing Rate, which is a good indicator of this behavior, as shown in Fig. 1.

The audio file is partitioned into segments of 2.04 seconds; each of them is composed of 150 consecutive non-overlapping frames. These values allow a statistical significance of the frame number and, using a 22050 Hz sample frequency, each frame contains 300 samples, which is an adequate trade-off between the quasi-stationary properties of the signal and a sufficient length to evaluate the ZCR.

For every frame, the value of the ZCR is calculated using the definition given in [5].

These 150 values of the ZCR are then used to estimate the following statistical measures:

- *Variance*: which indicates the dispersion with respect to the mean value;
- *Third order moment*: which indicates the degree of skewness with respect to the mean value;
- Difference between the number of ZCR samples, which are above and below the mean value.

Each segment of 2.04 seconds is thus associated with a 3-dimensional vector.

To achieve the separation between speech and music using a computationally efficient implementation, a multivariate Gaussian classifier has been used.

At the end of this step we obtain a set of consecutive segments labeled like speech or no-speech.

The next regularization step is justified by an empirical observation: the probability to observe a single segment of speech surrounded of music segments is very low, and viceversa. Therefore, a simple regularization procedure is applied to properly set the labels of these spurious segments.

The boundaries between segments of different classes are placed in fixed positions, inherently to the nature of the ZCR algorithm. Obviously these boundaries aren't placed in a sharp manner, thus a fine-level analysis of the segments across the boundaries is needed to determinate a sharp placement of them. In particular, the ZCR values of the neighboring segments are processed to identify the exact position of the transition between speech and music signal. A new signal is obtained from these ZCR values, applying this function

$$y[n]=\frac{1}{P}\sum_{m=n-P/2}^{n+P/2}(x[m]-\bar{x}_n)^2 \quad \text{with} \quad P/2<n<300-P/2$$

Where x[n] is the n-th ZCR value of the current segment, and $\bar{x}_n$ is defined as:

$$\bar{x}_n=\frac{1}{P}\sum_{m=n-p/2}^{n+P/2}x[m]$$

Therefore y[n] is an estimation of the ZCR variance in a short window. A low-pass filter is then applied to this signal to obtain a smoother version of it, and finally a peak extractor is used to identify the transition between speech and music.

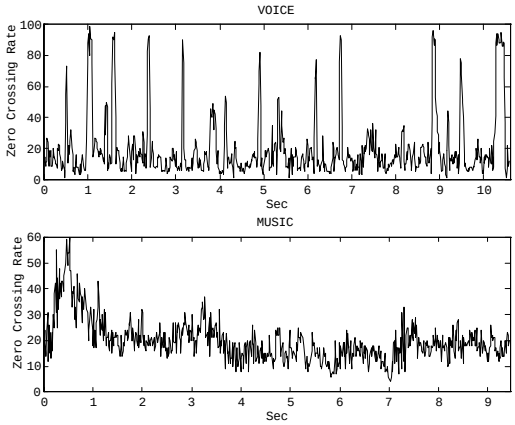


Figure 1. The ZCR behavior for voice and music segments.

4. NEURAL NETWORK CLASSIFIER

A Multi-Layer Perceptron (MLP) network [12] has been tailored to distinguish between music and speech. In multimedia applications mixed conditions must be managed, as music with a very rhythmic singer (i.e. rap song) or speech over music, as in advertising occurs. The MLP has been trained only by five kinds of audio traces, supposing other audio sources, as silence or noise, to be previously removed: pure music (class labeled as "Am"), melodic songs (class labeled as "Bm"), rhythmic songs (class labeled as "Cm"), pure speech (class labeled as "Av") and speech superimposed on music (class labeled as "Bv").

Eight features have been selected as the neural network inputs. These parameters have been computed considering 86 frames by 1024 points each (sampling frequency f_s=22050Hz), with a total observing time of about 4s. To allow a fine change detection, a circular

frame buffer has been provided and features p_j, in terms of mean value and standard deviation, are updated every 186 ms, corresponding to 4 frames f_i, as depicted in Fig. 2.

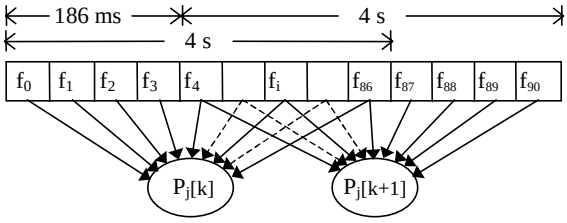


Figure 2. Features P_j updating frequency.

A short description of the eight selected features follows. Parameter P_1 is the spectral flux, as suggested by [13]. It indicates how rapidly changes the frequency spectrum, with particularly attention to the low frequencies (up to 2.5kHz) and it generally assumes higher values for speech.

Parameters P_2 and P_3 are related to the short-time energy [14]. Function E(n), with n=1 to 86, is computed as the sum of the square value of the previous 1024 signal samples. A fourth-order high-pass Chebyshev filter is applied with about 100Hz as the cutting frequency. Parameter P_2 is computed as the standard deviation of the absolute value of the resulting signal, and it is generally higher in speech. Parameter P_3 is the minimum of the short-time energy and it is generally lower in speech, due to the pauses that occur among words or syllables.

Parameters P_4 and P_5 are related to the cepstrum coefficients, as indicated in equation 1.

$$c(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\log\left|X(e^{jw})\right|e^{jwn}dw \qquad (1)$$

Cepstrum coefficients $c_j(n)$, suggested in [15] as good speech detectors, have been computed for each frame, then the mean value $c_\mu(n)$ and the standard deviation $c_\sigma(n)$ have been calculated and parameters P_4 and P_5 result as indicated in equation 2.

$$P_4=c_\mu(9)\cdot c_\mu(11)\cdot c_\mu(13),\ P_5=c_\sigma(2)\cdot c_\sigma(5)\cdot c_\sigma(9)\cdot c_\sigma(12) \qquad (2)$$

Parameter P_6 is related to the spectral barycentre, as music is generally more sensible in the low frequencies. In particular, the first-order-generalized momentum (barycentre) is computed starting from the spectrum module of each frame. Parameter P_6 is the product of the mean value by the standard deviation computed by the 86 values of barycentre. In fact, due to the speech discontinuity, standard deviation makes this parameter more distinctive.

Parameter P_7 is related to the ratio of the high-frequency power spectrum (7.5kHz<f<11kHz) to the whole power spectrum. The speech spectrum is usually considered up to 4kHz, but the lowest limit has been increased to consider signals with speech over music. To consider the speech discontinuity and increase the discrimination between speech and music, P_7 is the ratio of the mean value to the standard deviation obtained by the 86 values of the relative high-frequency power spectrum. Parameter P_8 is the syllabic frequency computed starting from the short-time energy calculated on 256 samples (≈12ms) instead of 1024. A 5-taps median filter has filtered this signal and P_8 is the number of peaks detected in 4s. As it is known [18], music should present a greater number of peaks.

To train and preliminarily test features and the MLP, a set of about 400 4s-long audio samples have been considered belonging to the five classes labeled as Am, Bm, Cm, Av, Bv and equally distributed between speech (Av, Bv) and music (Am, Bm, Cm). The discrimination power of the selected features has been firstly evaluated by computing index α, defined by equation (3), for each feature P_j, with j=1 to 8, where μ_m and σ_m are respectively the mean value and standard deviation of parameter P_j for music samples, and μ_v and σ_v are the same for speech. α-values between 0.7 and 1 result for the selected features.

$$\alpha=\left|\frac{\mu_m-\mu_v}{\sigma_m+\sigma_v}\right| \qquad (3)$$

The selected MLP has eight input, corresponding to the normalized features $P_1 \div P_8$, fifteen hidden neurons, five output neuron, corresponding to the five considered classes, and uses normalized sigmoid activation function. The 400 4s-long audio samples have been divided in three sets: training, validation and test. Each sample is formatted as $\{P_1 \div P_8, P_{Av}, P_{Bv}, P_{Am}, P_{Bm}, P_{Cm}\}$, where P_{Av} is the probability that sample belongs to class Av. The goal is to distinguish between speech and music and not to identify the class; for this reason target has been assigned with "1" to the selected class, "0" to the farest class, a value between 0.8 and 0.9 to the similar classes and a value between 0.1 and 0.2 to the other classes. For instance if a sample of Bm is considered, that is melodic songs, P_{Bm}=1, P_{Am}= P_{Cm}= 0.8 because music is dominant, P_{Bv} = 0.2 because it is anyway a mix of music and voice, and P_{Av}= 0.1, because the selected sample contains voice. If a pure music sample is considered (class Am), P_{Am}=1, P_{Bm}= $_{Cm}$= 0.8 because music is dominant, P_{Bv} = 0.1 because it is anyway a mix of music and voice, and P_{Av}= 0, because pure speech is the farest class. In fact, classifying the speech over music as speech inclines the

MLP to classify as speech some rhythmic songs: by adjusting the sample target it is possible to incline to one side or another the MLP response. In this application we suppose to discriminate between speech and music to successively identify particular words from speech, so a light preference to speech is acceptable. The MLP has been trained by Matlab tools using the Levenberg-Marquardt method with a starting μ value equal to 1000. The decision algorithm is depicted in Fig. 3. The mean error related to the 400 samples is 4%.

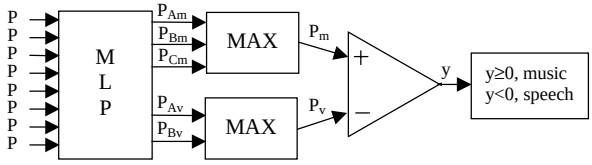


Figure 3. The decision algorithm.

5. SIMULATION RESULTS

The proposed algorithms have been tested by computer simulations to measure the classification performance. The tests carried out can be divided in two categories: the first one is about the misclassification errors, while the second one is about the precision in music-speech and speech-music change detection.

Considering the misclassification errors, we defined three parameters as follow:

- MER (Music Error Rate): it represents the ratio between the total duration of the segments misclassified, and the total duration of the test file.
- SER (Speech Error Rate): it represents the ratio between the total duration of the segments misclassified, and the total duration of the test file.
- TER (Total Error Rate): it represents the ratio between the total duration of the segments misclassified in the wrong category (both music and speech), and the total duration of the test file.

The "generation" of the test files was carried manually, i.e., each file is composed of many pieces of different types of audio (different speakers over different enviromental noise, different kinds of music as classical, pop, rap, funky,…) concatenated in order to have five minutes segment of speech followed by five minutes segment of music, and so on, for a total duration of 30 minutes.
All the content of this file has been recorded from a FM radio station, and it has been sampled at a frequency of 22050 Hz, with a 16 bit uniform quantization.
The classification results for both the proposed methods are shown in Table 1.

	MER	**SER**	**TER**
MLP	11.62%	0.17%	6.0%
ZB	29,3%	6,23%	17.7%

Table 1. Classification results of the proposed algorithms (MLP: Multi Layer Perceptron; ZB: ZCR with Bayesian Classifier).

From the analysis of the simulation results, we can see that, the MLP method gives better results compared to the ZB one, having a lower error rate both in music and speech.
Moreover, both the methods show the worst performance in the classification of the music segments, i.e., many segments of music are classified as speech than viceversa.
For a better understanding of these results, it can be useful take a look to the Fig. 4.

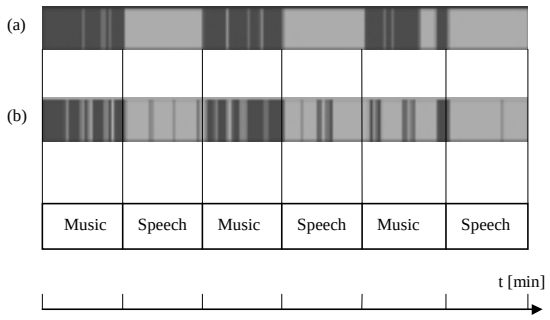


Figure 4. Graphical display of the classification results (a: MLP, b: ZB).

In the first row are shown the classification results of the MLP algorithm, where the white intervals are the segments classified as speech and the black ones are the segments classified as music.
The second row shows the classification results obtained using the ZB algorithm.

From the figure, it appears clearly that the worst classifications are carried out in the third music segment, between the minutes 20 and 25. The explanation is that these pieces of music are styles containing strong voiced components, under a weak music component (rap and funky).
The neural network makes a mistake only with the rap song, while the ZB approach performs a misclassification with the funky song too.
This is due mainly to these reasons:

- The MLP has been trained to recognize also music

with a voiced component, and it gets wrong only if the voiced component is too rhythmic (e.g., rap song in our case). On the other hand, the Bayesian classifier used in the ZB approach does not take in account cases with mixed component (music and voice), and therefore in this case the classification results are significantly affected by the relative strongness of the spurious components.

- Furthermore, the ZB approach, that uses very few parameters, is inerently not able to discriminate between pure speech and speech with music background, while the MLP network, which uses more features, is able to make it.

Considering the precision of music-speech and speech-music change detection, we measured the distance between the correct point in the time scale when a change occurred, and the nearest change point automatically extracted from the proposed algorithms.
In our specific test set, we have only five changes, and we have measured the maximum, minimum and the mean interval between the real change and the extracted one. The results are shown in Table 2, where PS2M (Precision Speech to Music) is the error in speech to music change detection, and PM2S (Precision Music to Speech) is the error in music to speech change detection.

	PM2S	**PS2M**
Min	0.56	0.19
Mean	1.30	1.53
Max	1.49	2.98

(a)

	PM2S	**PS2M**
Min	0.56	12.28
Mean	1.30	14.51
Max	2.79	16.74

(b)

Table 2. MLP (a), and ZB (b) change detection results expressed in seconds.

Also in this case, the MLP obtain better performance than the ZB.

6. CONCLUSION

In this paper we have proposed and compared two different algorithms for audio classification into speech and music. The first method is based mainly on ZCR and Bayesian Classification (ZB), and is very simple from the computational point of view.

The second approach uses Multi Layer Perceptron (MLP), and considers more features, requiring therefore more computations. The two algorithms have been tested to measure its classification performance in terms of misclassification errors and precision in music-speech change detection. Both the proposed algorithms give good results, even if the MLP shows the best performance.

REFERENCES

[1] MPEG Requirement Group. MPEG-7: Overview of the MPEG-7 Standard. ISO/IEC JTC1/SC29/WG11 N3752, FRANCE, Oct. 1998.

[2] C. Saraceno, R. Leonardi, “Indexing audio-visual databases through a joint audio and video processing”, International Journal of Imaging Systems and Technology, no. 9, vol. 5, pp. 320-331, Oct. 1998.

[3] N. Adami, A. Bugatti, R. Leonardi, P. Migliorati, L. Rossi, “Describing Multimedia Documents in Natural and Semantic-Driven Ordered Hierarchies”, Proc. ICASSP'2000, pp. 2023-2026, Istanbul, Turkey, 5-9 June 2000.

[4] A. Bugatti, R. Leonardi, L. A. Rossi, “A video indexing approach based on audio classification”, Proc. VLBV´99, pp. 75-78, Kyoto, Japan, 29-30 Oct. 1999.

[5] J. Saunders, “Real Time discrimination of broadcast music/speech”, In Proc. ICASSP’96, pp. 993-996, 1996.

[6] E. Scheirer, M. Slaney, “Construction and evaluation of a robust multifeature speech/music discriminator”, *In Proc. ICASSP’97*, 1997.

[7] J. Foote, “A similarity measure for automatic audio classification”, In Proc. AAAI'97 Spring Symposium on Intelligent Integration and Use of Text, Image, Video and Audio Corpora, 1997.

[8] C. Saraceno. *Content-based representation and analysis of video sequences by joint audio and visual characterization*. PhD thesis, Brescia, 1998.

[9] T. Zhang, C. C. Jay Kuo, “Content-based classification and retrieval of audio”, SPIE Conference on Advanced Signal Processing Algorithms, Architectures and Implementations, 1998.

[10] Z. Liu, J. Huang, Y. Wang, T. Chen, “Audio features extraction and analysis for scene classification”, Workshop on Multimedia Signal Processing, 1997.

[11] J.L. Gauvain, L. Lamel, G. Adda, “Partitioning and Transcription of Broadcast Data”, ICSLP, pp. 1335-1338, 1998.

[12] S. Haykin, “Neural Networks, a comprehensive foundation”, Prentice Hall, 1999.

[13] E. Scheirer, M. Slaney, “Construction and evaluation of a robust multifeature speech/music discriminator”, Proc. of the 1997 ICASSP, Munich, Germany, April 21-24, 1997.

[14] L.R. Rabiner, R.W. Shafer, “Digital processing of speech signals”, Prentice Hall.

[15] L.R. Rabiner, “Fundamental of speech recognition”, Prentice Hall, 1999.

A Novel Hexagonal Search Algorithm for Fast Block Matching Motion Estimation

Anastasios HAMOSFAKIDIS[1] and Yakup PAKER[2]

Dept of Computer Science, Queen Mary College, University of London, Mile End Road, E1-4NS, UK

Abstract—Based on real world image sequence characteristics of center-biased motion vector distribution, a Hexagonal (HS) algorithm with center-biased checking point pattern for fast block motion estimation is proposed. The HS is compared with full search (FS), four-step search (4SS), new three-step search (NTSS), and recently proposed diamond search (DS) methods. Experimental results show that the proposed technique provides competitive performance with reduced computational complexity.

I. INTRODUCTION

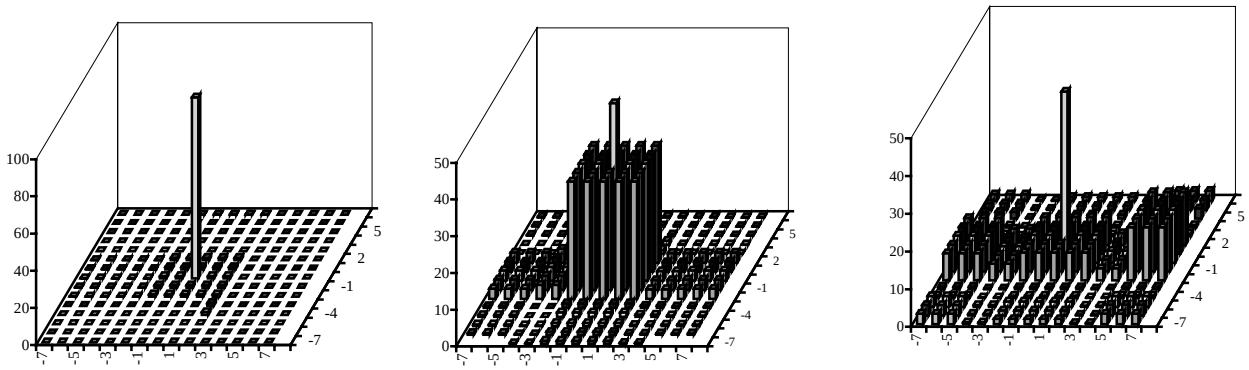
Motion compensated video coding, which predicts current frame from previous frames, has been widely used to exploit the temporal redundancy between consecutive frames. Motion estimation plays an important role in such an interframe predictive video coding system. Among different types of motion estimation algorithms, the block matching technique has been adopted in many compression standards, such as H.261 [1], MPEG-1 [2], MPEG-2 [3], and MPEG-4 [4]. In block matching, video data (frames or VOPs) are divided into blocks and one motion vector (MV) is associated with each block. For each block of the current frame/VOP a MV is derived which points to the best matching block of the previous (reference) frame/VOP. Then the best match block is used as the predictor for the current block.

The full search (FS) block matching algorithm is the simplest but the most compute intensive solution as it provides the optimal solution by matching all the possible displaced blocks within a given search range in the reference frame/VOP. In order to speed up MV derivation many fast block matching motion estimation (BMME) algorithms have been proposed such as three search step (3SS) [5], one at a time search (OTS) [6], four step search (4SS) [7], new three step search [8], and diamond search [9].
The objective of these fast BMME algorithms is to find the MV that minimizes the image error by reducing the number of checking points within the search window. Of these, 3SS and OTS algorithms are known to have the tendency to be trapped into a local minimum, thereby degrading performance. Based on the analysis of the above mentioned fast BMMEs and a study of MV distribution of real-world test video sequences a new hexagonal search (HS) algorithm is proposed in this paper. In the next section, we introduce the design motivation of the proposed algorithm. The algorithm description and simulation results are presented in Sections III and IV respectively. Finally, we give conclusions in Section V.

II. DESIGN MOTIVATION

[1] anastasi@dcs.qmw.ac.uk
[2] paker@dcs.qmw.ac.uk

Experimental results [7][9] have shown that the block motion field of real world video sequences is usually smooth, and varies slowly. That leads to a center-biased global minimum MV distribution instead of a uniform distribution. This can be observed from the MV distribution based on the FS algorithm for the first 100 frames of two test video sequences: the well-known "News", and the "Rallycross". The object-based MPEG-4 video standard segments the frames of video sequences into video objects (VOs). We have studied the MV distribution of News 0, News 1 VOs, and Rallycross. For the News 0 (background object) sequence, nearly all the blocks (97.55%) can be considered stationary, Figure 1a. For the News 1 (dancer) sequence of faster motion and camera zooming, the MV distribution is still highly center-biased: 48.31% found at the center of the search area, and 80% of them are enclosed in a central 5x5 area, Figure 1b. For the non segmented fast motion sequence "Rallycross" 52.76 % of its MVs are enclosed in a central 5x5 area and 61.16% are located in a central search 9x9 area, Figure 1c.



(a) (b) (c)

Figure 1: Motion vector distribution for (a) "News 0", (b) "News 1", and (c) Rallycross sequences.

Since the distribution of the global minimum point in real world video is centered at zero, fast BMME algorithms have been developed using center-biased checking point patterns. Analytically, the NTSS employs a center-biased checking point pattern combined with a halfway-stop technique, and achieves better performance than TSS. Using a moderate search pattern with fixed size of 5*5 the 4SS obtains a performance that is similar to NTSS. DS employs two diamond search patterns size of 9x9 and 3x3 respectively, which do not cover edge points of the search area. It becomes clear that the shape and the size of the above mentioned search patterns jointly determine not only the image quality (error performance) but also the computational complexity of fast BMMEs.

Based on the observation that global minimum distribution is centralized in real world sequences, the search points positions included to a zero-centered search 5x5 area are the most appropriate ones to be chosen to compose the search pattern. This choice is quite crucial in terms of algorithm's complexity and performance. This is the motivation behind the new hexagonal search (HS) algorithm proposed in here.

III. THE HEXAGONAL SEARCH ALGORITHM (HS)

Since motion vectors are not evenly distributed in the search area in fact most of them are located inside a centre-biased window of size 9x9, the HS patterns are designed to take into account the following:

(i) *reduced computational complexity*: the point where the minimum block distortion (MBD) occurs should be tracked using a small number of checking points, which cover a significant portion of the centre-biased search window, and

(ii) *search patterns shape:* when the MBD point is located, the search pattern has to be shaped in such a way that allows a refined search which covers all searching points around the MBD point in order to derive the MBD point for the best matching block.

As shown in figure 2, the HS algorithm utilises a centre-biased search pattern of seven checking points, out of which six points surround the centre one to compose a hexagon (Step 1). The hexagon points are checked and the centre of the hexagonal search window is then shifted to the point with minimum block distortion. The search pattern and its size, for the next two steps of the HS, depend on the location of the MBD points. If the MBD point is found at the center of the hexagonal pattern, the search proceeds to the final step (Step 3), with a smaller search pattern for a refinement search. Otherwise, the hexagonal search pattern is applied repeatedly until the MBD point is found at the center of the hexagon (Step 2). When the final step (Step 3) is reached, the search pattern is changed from hexagonal to a star, figure 2 (b) with a variable number of search points, best case 4 and worst case 6. For edge points of the search area the hexagonal search pattern (step 2) is modified, figure 3. The HS algorithm is summarised as follows:

Step 1: The initial hexagonal pattern is centered at the origin of the search window and the seven checking points (●) of the hexagon are tested, figure 2 (a). If the MBD point is found at the center position then go to Step 3, otherwise go to Step 2.

Step 2: The MBD point found in the previous search step is re-positioned as the center point to form a new hexagon. If the new MBD point obtained is located at the centre position, go to Step 3; otherwise, recursively repeat Step 2. The hexagonal pattern is modified on the borders of the search area in order to cover the edge points. Figure 3 (a) presents all the possible shapes of the hexagonal pattern when it reaches the left/right or the up/down borders of the search area. More precisely, there are two different scenarios when the pattern reaches the top or down borders. The first scenario is the centre point of the pattern, when it is shifted towards the up or down borders, to be on the border. In this case the new hexagonal pattern employs 4 checking points. The other scenario is when the checking points of the shifted hexagonal pattern are out of the borders of the search area, in this case the shifted pattern has 6 checking points. Similarly there are two cases when the shifted hexagonal pattern reaches the right or left borders of the search area. One case is when the centre of the shifted pattern is on the border of the search area, and the other case is where checking points of the shifted pattern lie outside the borders. In both cases the modified pattern has 5 checking points.

Step 3: Switch the search pattern from hexagon to star (♦), figure 2 (a).

There are two different star patterns that are employed for different locations of the MBD point in the search area

- The MBD point is not an edge point. The initial star pattern, figure 2 (a) is centered to the MBD point of the hexagonal pattern and its four checking points are tested. If the new MBD point calculated for the star pattern is located at the centre then this point is the final solution for the motion vector and the search stops. Otherwise, if the new MBD point is one of the other points of the initial star then its neighbouring points, excluding the central star point, are checked figure 2 (b). The new derived MBD point is the final solution of the MV since it generates the smallest MBD in the pattern.
- The MBD point is an edge point. The initial star pattern, adjusted to three checking points, is centered on the MBD of the hexagonal pattern, and its three checking points are tested, figure 3 (b). If the MBD point is at the centre of the modified star the search stops, otherwise the neighbouring points of the MBD, excluding the central point, are examined. The new derived MBD point is the final solution of the motion vector that points to the best matching block.

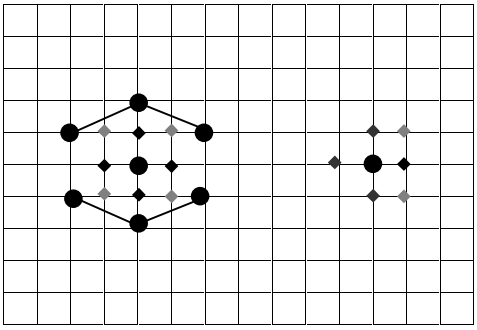


Figure 2: (a) Hexagon and initial star patterns, neighbouring points of the initial star are shown by grey colour, (b) Expanded star pattern for no edge points.

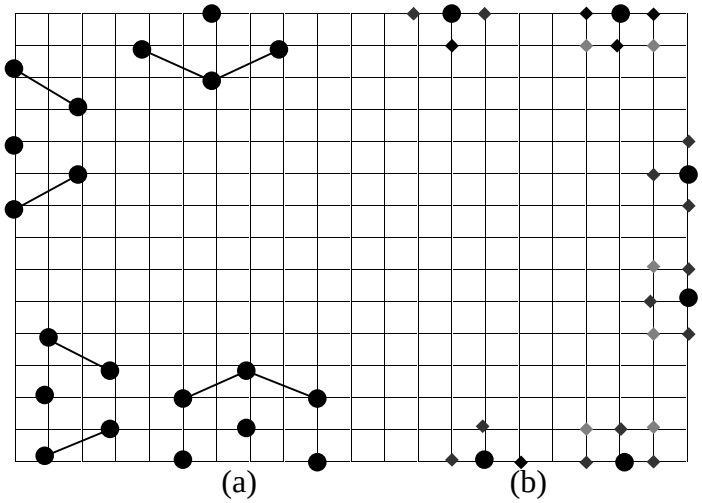


Figure 3: (a) all possible shapes of the hexagonal pattern when it reaches the left/right or the up/down limits of the search area, and (b) possible star shapes (initial and expanded) for a down/up or right edge MBD points.

Note that the checking points of the hexagon search pattern are partially overlapping when Step 2 is repeated. Only three checking points need to be calculated in the new pattern. In addition, at Step 3 when the search pattern changes from hexagon to star, three, four, or six points of the star need to be calculated, depended on star's MBD point location.. Figure 4 presents an example of how the HS derives a motion vector from the borders of the search area, MV (-7,7).

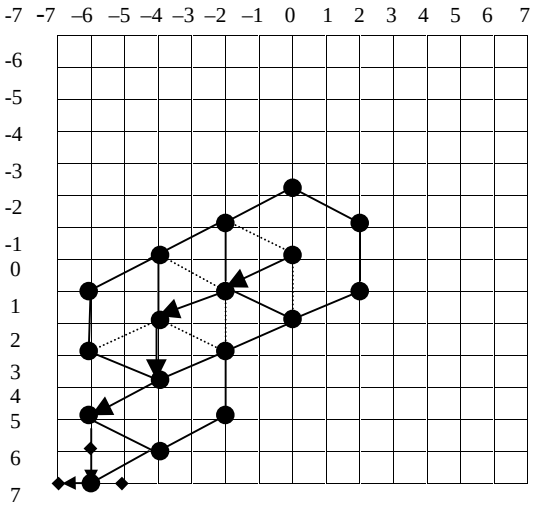


Figure 4: HS search path for MV (-7,7)

IV. SIMULATIONS

In our simulations, the BDM is defined to be the sum of absolute difference (SAD). For a given displacement (x, y), SAD is defined as $SAD(x, y) = \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} | I_k(m, n) - I_{k-1}(m+x, n+y) | *(alpha_k \neq 0)$ [10][11], $I_k(x,y)$ is the pixel intensity (luminance or Y component) at location (x,y) in the *k*-th frame/VOP, and $alpha_k$ refers to the current VOP at time instance k and contains the information which of the pixels form an object ($alpha_k > 0$) and which are outside the object ($alpha_k = 0$). The block size is fixed at *N*x*N* with *N*=16, and the maximum motion in row and column is assumed to be ±7. The first 100 frames of "News", and "Rallycross" video sequence are used. Analytically two fixed size VOs of the News sequence, News 0 (slow motion), and News1 (faster motion and zooming), and the "Rallycross" fast motion video sequence. We use the peak signal to noise ratio (PSNR) as the measure of performance. The PSNR is an image quality metric where larger values of it translated to better quality, $PSNR = \frac{20 \log_{10} 255}{\sqrt{MSE}}$ (MSE is the mean squared error). The required average number of search points for each block is used as the measure of computational complexity. Each video sequence is processed by five algorithms: full search (FS), diamond search (DS), four step search (4SS), new three-step search (NTSS), and the proposed hexagon search (HS). The degree of computational complexity of each algorithm with respect to full search algorithm is calculated. The simulation results are shown in Table I.

TABLE I
AVERAGE SEARCH POINTS PER MOTION VECTOR ESTIMATION AND PSNR FOR THE FIRST 100 FRAMES

Algorithms	News 0			News 1			Rallycross			Average Complexity
	Av. SP per MV	Complexity	Average PSNR	Av. SP per MV	Complexity	Average PSNR	Av. SP per MV	Complexity	Average PSNR	
FS	225	100%	35.91	225	100%	32.04	225	100%	34.15	100%
4SS	17.03	7.57%	35.89	18.37	8.16%	32.01	20.67	9.18%	34.13	8.30%
NTSS	17.05	7.58%	35.89	18.96	8.42%	32.00	21.01	9.34%	34.12	8.45%
DS	13.05	6%	35.9	15.25	6.77%	32.00	17.99	7.99%	34.14	6.92%
HS	11.06	4.91%	35.9	12.67	5.63%	32.02	14.44	6.41%	34.14	5.65%

The simulation shows that HS performs better than DS, 4SS and NTSS. An interesting observation is that for sequences of large MV distribution (Rallycross), HS outperforms all the others fast search algorithms. For instance for the Rallycross video trailer the HS computational complexity is 6.41% while DS, 4SS, NTSS complexities are 7.99%, 9.18%, and 9.34% respectively with similar performance to FS in terms of PSNR, Table I.

V. CONCLUSIONS

In this paper, the HS algorithm is proposed to perform block motion estimation in video coding. Based on the observations that global minimum distribution is centralized in real world video sequences and shape and size of search patterns determine not only the performance but also computational complexity of fast BMMEs, the HS employs a center-biased hexagon search pattern. The HS also employs the concept of variable size search patterns that allow it to cover all the search area in a small number of steps. Experimental results show that the proposed HS outperforms DS, 4SS, and NTSS search algorithms for fast

motion video sequences, having always better average computational complexity (slow, medium, fast motion) with similar performance to FS in terms of PSNR.

REFERENCES

[1] International Telecommunication Union, “Video codec for audiovisual services at px64 kbits”, ITU-T Recommendation H.261, March 1993

[2] ISO/IEC JTC1/SC29/WG11, “ISO IEC CD 11172: Information Technology”, MPEG-1 Committee Draft, December 1991

[3] ISO/IEC JTC1/SC29/WG11, “ISO IEC CD 13818: Information Technology”, MPEG-2 Committee Draft, December 1993

[4] “Special Issue on MPEG-4”, IEEE Transactions on Circuits and Systems for Video Technology, vol 7, no 1, February 1997

[5] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion-compensated interframe coding for video conferencing”, in Proc. NTC81, New Orleans, LA, pp. C9.6.1-9.6.5, Nov 1981.

[6] R. Srinivasan and K.Rao “Predictive coding based on efficient motion estimation”, IEEE Transactions on Communications, vol. COM-33, pp. 888-896, August 1985

[7] L. Po, and W. Ma , “A novel four-step search algorithm for fast block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 313-317, June 1996

[8] R. Li, B. Zeng, L. Liou, “A New Three-Step Search Algorithm for Block Motion Estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 4, pp. 438-442, August 1994.

[9] S.Zhu, and K. Ma “A new diamond search algorithm for fast block matching motion estimation”, International Conference on Information, Communications and Signal Processing, ICICS’97, pp. 292-296, 9-12 September 1997, Singapore.

[10] T. Ebrahimi “MPEG-4 video verification model version 8.0”, no N1796, Stockholm MPEG-4 meeting, July 1997

[11] “MPEG-4 simulation software”, ISO/IEC N2205, Tokyo MPEG-4 meeting, March 1998

USING MPEG-7 AT THE CONSUMER TERMINAL IN BROADCASTING

Alan Pearmain
Electronic Engineering Department, Queen Mary, University of London,
Mile End Road, London E1 4NS, ENGLAND
Tel: +44 20 7882 5342; fax: +44 20 7882 7997
e-mail: alan.pearmain@elec.qmw.ac.uk

Mounia Lalmas, Ekaterina Moutogianni Damien Papworth, Pat Healey and Thomas Rölleke
Computer Science Department, Queen Mary, University of London,
Mile End Road, London E1 4NS, ENGLAND

ABSTRACT

The IST SAMBITS (System for Advanced Multimedia Broadcast and IT Services) project is using Digital Video Broadcasting (DVB), MPEG-4 and MPEG-7 in a studio production and multimedia terminal system to integrate broadcast data and Internet data. This involves using data delivery over multiple paths and the use of a back channel for interaction. MPEG-7 is being used to identify programme content and to construct queries to allow users to identify and retrieve interesting related content. Searching for content is being carried out using the HySpirit search engine.

1 INTRODUCTION

SAMBITS is a European Union IST project investigating ways in which broadcasting and the internet can work together. The project is working on studio systems for producing content that allow a broadcaster to link broadcasting and the internet and also working on terminals capable of displaying this content in a way that is accessible to ordinary users [1].

Normal MPEG-2 broadcast content is sent by standard DVB techniques, but this is linked to extra content that consists of MPEG-4 audio-video sequences and HTML pages. MPEG-2 and MPEG-4 multimedia information has MPEG-7 [2] metadata added at the studio which describes certain features of the content. The extra MPEG-4 and HTML content may be sent over the MPEG-2 transport stream in private sections or it may be sent over the internet.

The terminal is based on the Multimedia Home Platform (MHP) reference software running on a set-top box. MHP currently only supports MPEG-2, so the project is adding software to support MPEG-4 and MEPG-7, storage of multimedia content and searching of multimedia content. It is intended that the user will be able to access this content with a system that is very similar to a standard set-top box and television with a remote control.

The SAMBITS project has twelve partners and one of the main contributions of Queen Mary is in the terminal: the MPEG-7 descriptors, information retrieval and user interface. There will be a demonstration of the project at IBC2001 in Amsterdam.

2 BACKGROUND

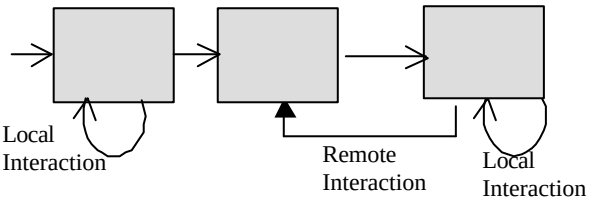


Figure 1 The SAMBITS system

The complete system that is being developed is shown in Figure 1. The studio system involves the development of various authoring and visualization tools. Standard equipment is being used for the broadcast and internet servers and the terminal development is based on the Siemens ACTIVY set-top box.

Some of the functions that will be available in the terminal are:

- Instant access to additional content, which may be provided via DVB or via the Internet.
- Access to information about the current programme.
- Searching for additional information either using metadata from the current programme or using a stored user profile.

The whole system provides a platform for investigating how MPEG-7 descriptors can be used at the consumer end in a broadcasting environment. The first problem was to choose a suitable set of descriptors. The descriptors that will be useful to a user will be high-level descriptions of the content. The studio will also include lower-level descriptors such as the percentage of different colours in a scene or camera information, but these would not be useful at the terminal.

User interaction is limited to remote control buttons, rather than a keyboard as many television users do not feel comfortable having to use a keyboard and keyboards are bulky and relatively expensive. This produces some challenges for the user interface design, particularly in the construction of queries.

The user will have the option whether or not to display the MPEG-7 data that is associated with the current programme via an "Info" button on the remote control. Searches are constructed based on the MPEG-7

metadata available for the current programme. The retrieval engine uses HySpirit (www.hyspirit.de) a retrieval framework based on probabilistic relational algebra [3].

3 THE USER INTERFACE

The overall architecture of the MPEG-7 user interface is shown in Figure 2.

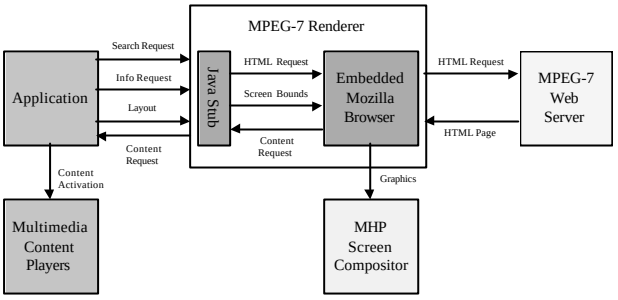


Figure 2 Display of Information in the terminal

The MPEG-7 user interface uses an integrated browser based on the Mozilla HTML browser. The MPEG-7 information is transformed from XML to HTML using style sheets, and the HTML is then rendered by the embedded browser.

Additional controls for the MPEG-7 engine, such as searching for related material, are also placed in the generated HTML pages.

4 CONTENT DESCRIPTION

Many of the descriptors that appear in the MPEG-7 standard are of little interest to the end consumer of multimedia content, but provide useful information for content providers or are provided for control of ownership and access rights. An overview of the MPEG-7 descriptors that will be exploited at the terminal is shown in Table 1. The descriptors that are directly useful to the user consist of a textual and a structured description of the nature of the segment, creation information and classification information, so these are the ones that will be displayed on the terminal.

The definition of descriptors within the MPEG-7 standard is still ongoing, but these descriptors are in the set that is currently the candidate for adoption in the standard. This list may also be updated to adapt to any change arising in the SAMBIT scenarios (see Section 7).

Figure 3 shows the general MPEG-7 part of the terminal. MPEG-7 data can be transported over the broadcast channel either as text or as a binary representation. The binary representation is still being developed within the standardisation process. If a binary form is used, it must first be decoded to the text description, which is an XML structure. An XSLT processor is then used, together with a style sheet, to produce a HTML version of the description. The HTML is sent to a local web server on the terminal. If the user requests the MPEG-7 data about the current programme, the HTML browser on the terminal is used to send a request to this local web server.

D- / DS-Name	Contained D-/DS-s
MediaInformation	
MediaTime	MediaRelTimePoint, MediaDuration
FreeTextAnnotation	
StructuredAnnotation	Who, Where, What
Creation	Title, Abstract, Creator, CreationCoordinates
Classification	Country, Language, Genre, Subject
RelatedMaterial	MediaType, MediaLocator
SegmentDecomposition	DecompositionType(temporal, spatial), Segment
UserPreferences	User Identifier, BrowsingPreferences,FilteringAndSearchPreferences

Table 1 Descriptors

The way in which the content data is displayed is shown in Figure 4. The bottom strip on the screen shows the control buttons that have different uses depending on the mode of the terminal. In the MPEG-7 information display mode the round circle button allows the data display to be turned on or off.

5 SEARCHING

Queries are constructed from MPEG-7 data for the current programme. An example of query construction is shown in Figure 5 Constructing a query

The user has asked for further information and has been presented with the information that is immediately available. He or she can select which of these items he or she wants to select with the up/down buttons on the remote control. Users are also presented with an option to extend the search. The search could then be extended either to the Internet server of the broadcaster or to the whole of the internet.

The query is formulated as an XML query and sent to the HySpirit search engine (www.hysirit.de). This is a retrieval framework based on a probabilistic extension of well-known database data models such as the relational model, the deductive model, and the object-oriented model for information retrieval purposes. HySpirit allows the capturing of content (e.g. terms), facts (e.g. authors) and structure (e.g. XML) in retrieving information from semi-structured and heterogeneous data sources.

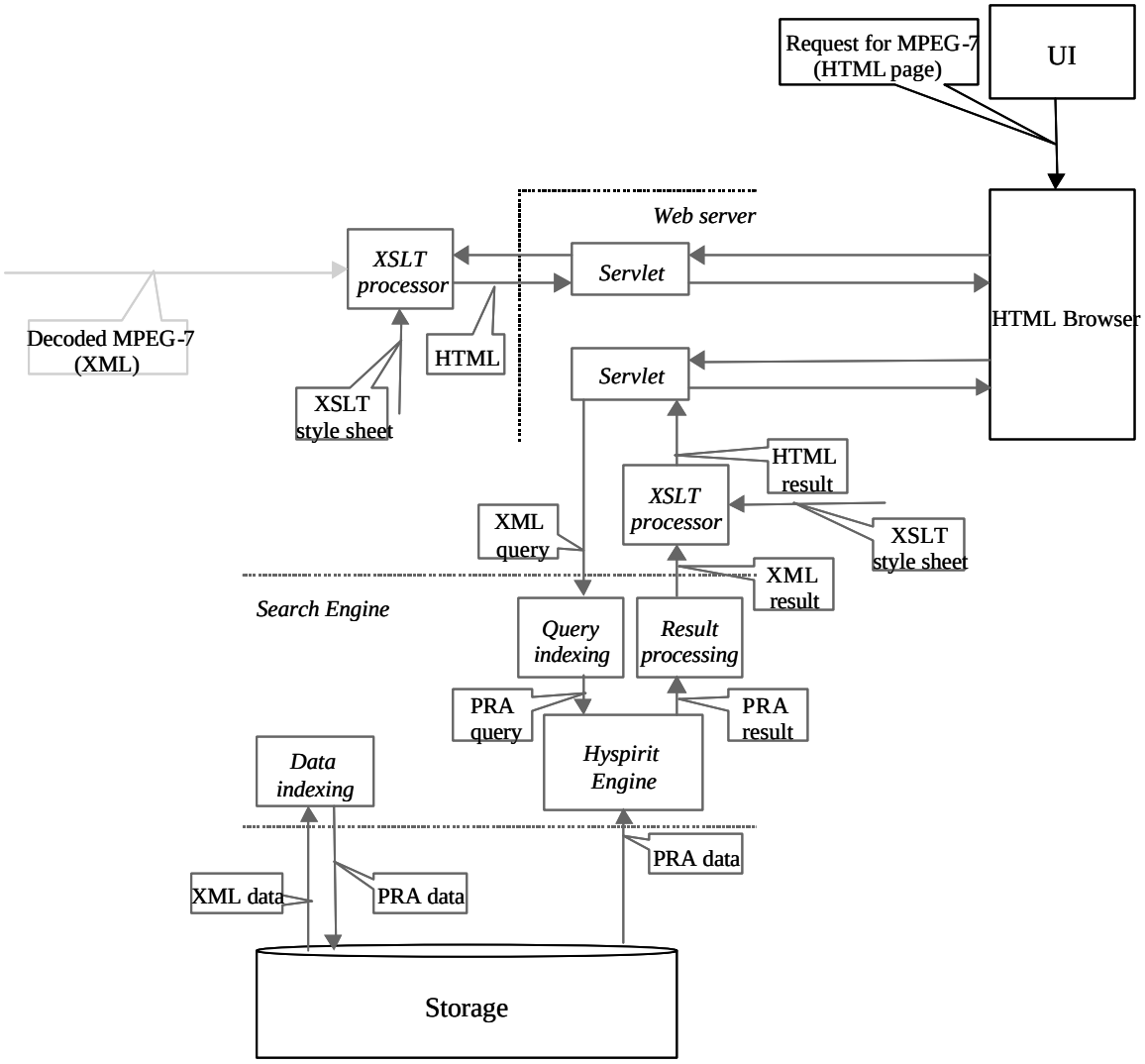


Figure 3 MPEG-7 processing and searching

Figure 4 Display of MPEG-7 data

Figure 3 shows the system for processing a query. The query is formed in the HTML browser and sent to the local web server. The query is then sent to the search engine as a XML query. An indexing module in the search engine converts the XML query to a Probabilistic Relational Algebra (PRA) query that is suitable for submission to HySpirit and HySpirit returns PRA results that are converted to XML. These are then processed with an XSLT style sheet to give the results in a rank order as HTML to send to the browser. Figure 6 shows the results of the search with the ranking as a percentage. We may present this ranking information is some alternative graphical form. A filter module for the system is not shown in Figure 3, but this will be included so that the results presented are based on a user profile (see below).

6 USER PREFERENCES

Users will be able to store a profile of their preferences and both the metadata about the current programme and the search results will be filtered according to this profile before display. Some preferences will relate to the type of data to be displayed, e.g. a user could select that he was not interested in place information or that he only wanted to see the two best match results from a search. Other options could be added about the interests of the user. A possible development would be to monitor user searches and requests to automatically build a user profile to filter results.

7 PROJECT DEMONSTRATION

The whole studio and terminal system being developed in SAMBITS will be demonstrated at IBC2001 in Amsterdam. At least two scenarios will be used in the demonstration, with the most complex scenario being based on the Eurovision song contest. This will allow metadata on singer, song title, country etc. to be provided and searches to be carried out to find more information about the singer, songs from previous years, information about the singer country of origin etc. and extra views of the contest, such as a backstage camera view. Some of this material will be available as MPEG-4 multimedia content via the object carousel and some will be the type of information that would normally be available from the internet.

Figure 5 Constructing a query

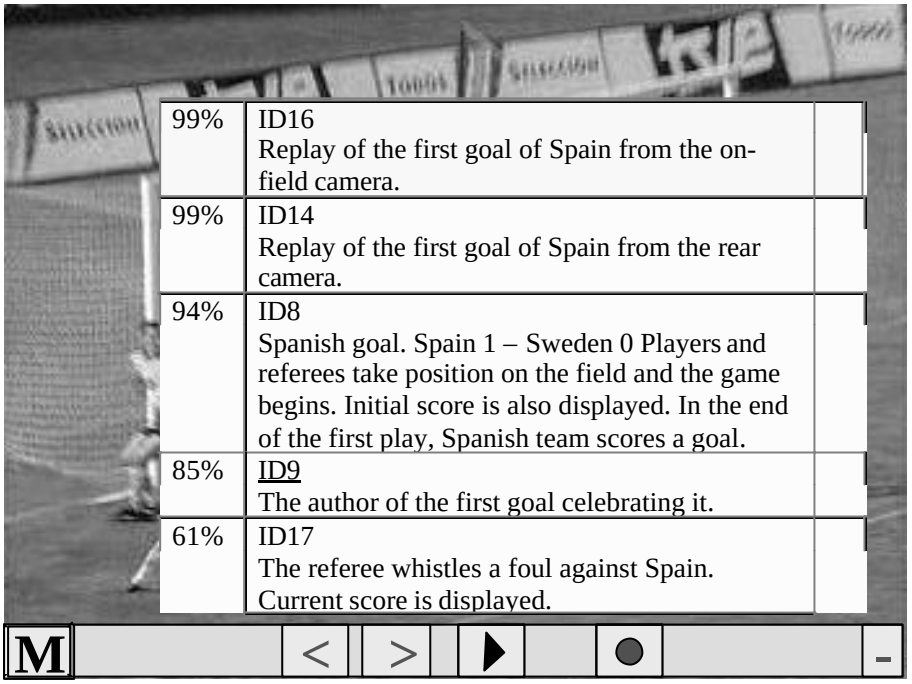


Figure 6 Display of search results

8 CONCLSUION

The SAMBITS project is developing a system for displaying MPEG-7 metadata associated with broadcast programme content and with related content that can be distributed via MPEG-2 transport streams or via the internet. The system also allows the construction of queries from this metadata using the set-top box remote control. The queries are submitted to the HySpirit search engine and the results are returned in rank order. Results are filtered according to user preferences. The system being developed should form an excellent platform for evaluating user reaction to these functions for integrating the internet with television.

REFERENCES

[1] P Healey, M Lalmas, E Moutogianni, Y Paker and A Pearmain. "Integrating internet and digital video broadcast data". Proceedings of 4th world multiconference on Systemics, Cybernetics and Informatics (SCI 2000), Orlando, Florida, July 23-26 2000 Vol.1 pp 624-627.

[2] ISO/IEC JTC 1/SC 29/WG 11 "N3465 MPEG7 Multimedia Description WD (Version 4.0)", 2000.

[3] N Fuhr and T Rölleke. "HySpirit – A probabilistic inference engine for hypermedia retrieval in large databases", Proceedings of International Conference on Extending Database Technology (EDBT), Valencia, Spain, 1998.

USER INTERFACE DESIGN FOR KEYFRAME-BASED BROWSING OF DIGITAL VIDEO

Hyowon Lee, Alan F. Smeaton, Noel Murphy, Noel O'Connor and Sean Marlow
Centre for Digital Video Processing
Dublin City University, Glasnevin, Dublin 9, Ireland.
Tel: +353 1 7005433; fax: +353 1 7005508
e-mail: murphyn@eeng.dcu.ie

ABSTRACT

In this paper we describe a structured approach for the development of user interfaces for the Físchlár video browsing system, a web-based system for recording, browsing and playback of TV programmes. The user interface to the system was originally designed for desktop use with a large screen and a mouse and we are currently developing versions suitable for mobile device (PDA) access to the system. We review a design framework for video browsing interface formats and some of the formats developed for desktop and PDA use, including interfaces for the Psion Revo and Compaq iPAQ PDAs. This work is driven by the need to investigate how best to include the user in the content specification and retrieval loop and how to find the various balance points between user interaction and system automation.

1 INTRODUCTION

The user interface is key to the acceptance of a media-related product in the marketplace. All the technology components can be in place and well integrated, but their effectiveness for an individual user will be unrealised if the user interface is unsuitable. For this reason, in the development and implementation of the Físchlár video browsing system, a web-based digital video retrieval system for TV programmes, we put substantial effort into a structured approach for the development of user interfaces. We have also rolled the system out to campus-resident users in order to get a broad spectrum of usage feedback. The user interface to the system was originally designed for desktop use with a large screen and a mouse. However, we are currently developing versions suitable for mobile device (PDA) access to the system to record and browse video content.

Físchlár is a web-based community-access digital video system with over 600 users within the campus environment in Dublin City University. The system allows the user to record broadcast TV programmes, and facilitates browsing and playback of the recorded programmes on a web browser. The user can easily browse eight terrestrial TV channel schedules for today and tomorrow, arranged in channel, genre, favourites or personalised recommender form. By simply clicking on a programme, they can set the recording. The system then encodes the programme in MPEG-1 format when the broadcast time comes. The encoded programme is subsequently subjected to automatic camera shot and scene boundary detection to extract representative keyframes. These are the visual medium for the user's interface with video retrieval functions. The web-based interface allows the user to select one of the several browsing methods we have developed to see the keyframes. Clicking on any of the keyframes will pop up a new window which starts streamed playback of the video from the clicked keyframe onwards. The video database system is capable of delivering about 150 independent video streams.

The Físchlár system is a testbed for our technology development, wherein any implemented techniques such as various shot/scene boundary detection algorithms [1], integration with a programme recommender system [2], mobile application for video browsing and playback [3], and various user interface ideas [4] are easily plugged in to the system and the outcomes visibly demonstrated to our current user base. Users of the system are an important element of our work, as they provide new ideas from their own, real, usage context. The Físchlár system is further described in O'Connor *et al* [5] and in [6].

In recognition of the diversity of users' preferences and task contexts, we have developed a design framework for video browsing interfaces that allows us to come up with many different formats of browsing interface. Using this framework we have implemented 8 different browser formats suitable for a desktop environment. The user chooses and uses these different interface formats according to their preferences, and according to their retrieval objectives, which vary from time to time — even within a single user session — and vary from person to person. As we are presently working on the use of mobile devices to access the Físchlár system, porting the systems browsing/playback features to mobile devices has become an important concern for us. In this paper we review the application of the browsing interface design

framework to desktop applications and some of the formats developed for desktop Físchlár use are described. We then show how the framework can also be applied to design suitable interfaces for handheld personal digital assistant (PDA) type devices with their small, touch-sensitive screen and mobile environment use. The resultant PDA video browsing implementations have highly interactive interfaces, but require relatively less visual attention and focusing and can be comfortably used in a mobile situation to browse the multimedia content.

Section 2 briefly explains the rationale for developing the design framework, and summarises the actual framework for keyframe-based browsing interfaces. In section 3, we show how it can be used to design a specific large-screen-and-mouse desktop browser. In section 4 we apply general interface design concerns for mobile devices to the design framework and demonstrate two example designs suitable for a PDA using the Psion Revo and the Compaq iPAQ. These, nevertheless, are primarily case studies. The design priority is to investigate how best to include the user in the content specification and retrieval loop, and how to find the various balance points between user interaction and system automation. Section 5 concludes with some future directions in video browsing interface design.

2 DESIGN FRAMEWORK FOR VIDEO BROWSING INTERFACES

One of the main problems in designing a user interface for a novel system such as a digital video browser is the lack of prior experience with any directly comparable type of human-computer interaction. Another is the fact that a single "optimised" interface cannot satisfy everybody, because people come to the system with different aptitudes, attitudes, preferences and task contexts. Furthermore, the current trend in technology is toward a diversification of devices using a single underlying system and sharing the same data, such as email software accessed from an office desktop PC, a PDA or a mobile phone. This results in the need to design different (though related) user interfaces for different devices that are suitable for different users and different contexts. To address these problems, there have been efforts to streamline and turn the fuzzy, unpredictable and ill-defined interface design approach into a more structured and formalised process, exemplified by "design space analysis" [7] and further adapted in various forms such as Stary [8]. In this approach, roughly the following steps are followed:

- analyse and identify important elements and alternatives in designing an interface, resulting in an exhaustive sets of possible design options, or design space,
- consider the particular environment where the interface in concern is to be used, and
- select a suitable set of options from the design space.

In this way, designing the functionality of an interface becomes less of an intuitive, artistic task and more of a concrete and simple decision-making process where the designer can come up with many different interfaces by selecting different combinations of options suitable for the target usage. A crucial part in this approach is the initial construction of the design space and the selection of the right set of options for the target usage. In designing video keyframe browsing interfaces for the Físchlár system, we identify three important design dimensions (layeredness, temporal orientation and spatial vs. temporal presentation) and several possible options or values for each dimension. The detail of the rationale leading to the selection of these particular dimensions and the selection of suitable values along each dimension are discussed in more detail in [9] and [10].

2.1 Layeredness

Keyframes extracted from a video programme are a useful way of providing an overview of the programme content, but the number of keyframes presented crucially affects the user's browsing. A large number of keyframes allows detailed browsing, but is unsuitable for quick browsing. The *layeredness* dimension is concerned with the different possible levels of detail or granularity of the keyframe set and the transition between different levels of granularity. Some of the typical options for this dimension are:

Single layer Provides only a single set of keyframes, whether very detailed or selective;

Multiple layer without navigational link Provides more than one set of keyframes in a browser, thus the user can select the granularity s/he wants in the browsing;

Multiple layer with navigational link Provides more than one set of keyframes, and the user can jump between different sets of keyframes while maintaining the current point of browsing.

2.2 Temporal Orientation

The keyframes extracted from video are an ordered set of images in time. Thus an important concern is what kind of *time* information, if any, should be provided to the user when browsing the keyframe set. Some of the typical options for this dimension are:

No time information Provides no explicit time information regarding each keyframe;

Absolute time Provides exact time information in numeric form (for example, time-stamping a keyframe with "15 minutes 30 seconds into the video");

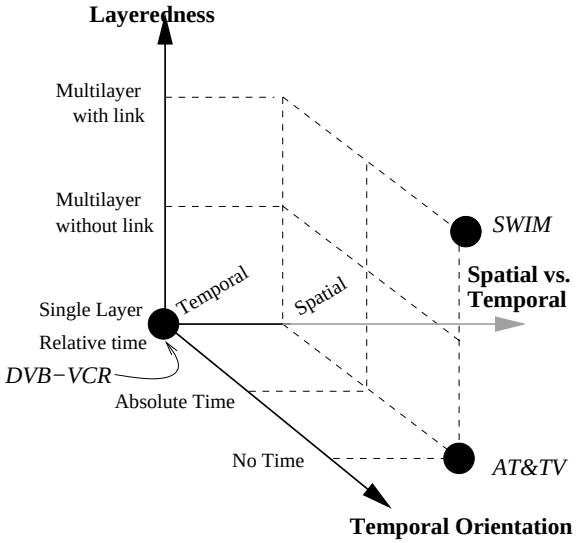


Figure 1: Diagram of a 3-D design space where each axis represents one of the design dimensions. The positions in this space of the SWIM hierarchical browser [11], DVB-VCR [12] and AT&TV [13] are shown.

Relative time Shows the time of the current browsing point in relation to the whole length of the video (for example, a timeline bar indicating the current viewing point).

2.3 Spatial vs. Temporal Presentation

There are two distinctive ways of presenting keyframes on the screen and the designer has to decide which one should be adopted for the interface in question:

Spatial presentation This displays many miniaturised keyframes side by side, allowing quick spatial browsing;

Temporal presentation This displays keyframes one by one in the manner of a slideshow.

Actually the two modes are not mutually exclusive and we have in some cases combined both modes in a single interface format. However, the different characteristics of the two modes become important in particular applications, so it is important to distinguish them.

3 SPECIFYING A DESKTOP BROWSING INTERFACE

The three dimensions and their typical options described above form a design space where the designer selects one (or more than one) option from each dimension. Because each design option represents a distinctive design decision in a dimension, different combinations of options result in distinctive browsing interfaces. This makes it possible to design all conceivable browsing interfaces within the constraints of the spatial dimensions.

The 3-dimensional space where each axis represents one of the dimensions described above is shown in Figure 1. In this space we can locate positions of several existing video browsing interfaces, for example, the SWIM hierarchical browser [11], DVB-VCR [12] and AT&TV [13]. These interfaces are then seen as choices within a systematic set of alternatives. By locating a selection of particular points in the space, it is possible to propose several further well-specified browsing interfaces that can then be evaluated against the design criteria for a particular target usage application or task.

An example of such an interface is the *Timeline Bar Browser* show in Figure 4. This involves presenting a fixed number of keyframes (24 shown) on one screen. Then as the user moves the mouse cursor over the timeline bar at the top, the screen of keyframes flips through to the next set of (24) keyframes. The timeline bar provides proper time orientation, as bar increments are proportional to the time covered by the set of 24 keyframes. Also, the "ToolTip" shows the exact time of the current screen. User feedback showed the initial implementation of this interface to be very easy to use, but suggested that it could be improved by having a "sticky" mouse pointer so that it is possible to concentrate on keyframes while moving the mouse over the timeline bar.

A selection of eight different interfaces implemented on the basis of this "space-filling" approach is illustrated by the icons in Figure 5. More details of the design criteria that each of these fulfils can be found in [9] and [10]. The Físchlár system for the campus-based "public" is currently running with a selection of five distinctive keyframe browsing interfaces. This reduced set is to avoid information overload of the users and to focus our user evaluations on a smaller of variables.

4 SPECIFYING A PDA BROWSING INTERFACE

In designing the browsing interfaces to the Físchlár system on a PDA, general guidelines and common sense can be used in this selection process. For example, on a small, low-resolution screen extensive spatial presentation is not suitable because each keyframe would be unrecognisably small.

However, apart from the physical limitations of PDA devices, the mobile PDA user environment is very different from that of the large-screen-and-mouse desktop. Well-established desktop GUIs are designed to keep the user looking at the screen with proprioceptive awareness of the mouse and/or fine hand-eye cursor control [14]. In the mobile environment (in a bus, on the street, on the metro) the user may be unable to keep focused on the screen, small visual details can easily be overlooked and only one hand may be available some of the time.

In our work to date we have designed PDA interfaces for the Psion Revo and Compaq iPAQ. The Revo has a 480×160-pixel landscape touch-sensitive screen with 16

shades of grey. Figure 6 shows one of our designed interfaces. The list of available TV video content is displayed on the right side of the screen, with a scroll bar for right thumb manipulation, while holding the device with the same hand. Below the keyframe from the selected programme displayed on the left side of the screen there are two buttons (previous/next) for the user to flip through keyframes one-by-one, using the left thumb while also holding the device with the left hand. A timeline bar beside the buttons shows the current point of browsing in relation to the whole programme. Automatically flipping through keyframes (true temporal presentation) would be possible, but it would force the user to keep concentrating on the screen. Requiring a high degree of interaction (repeated tapping on the previous and next buttons) should be okay with only two interaction objects (buttons) where these are always under the user's thumb. Note that it is possible interact with the device using only one hand at either stage of interaction. When both hands are available, the user can use the right thumb for scrolling and selecting a TV programme and the left thumb for flipping through the selected programmes keyframe content.

Another interface for the Revo is shown in Figure 7 below. This interface is designed for browsing the keyframes of a single programme, with multiple layers of keyframes available. With the two buttons on the right side (up/down buttons), the user can jump between 6 different layers, while the layer indicator beside the buttons shows the currently selected layer. The top layer has ten selected keyframes providing an overview of the whole programme; the bottom layer has the full camera shot-level set of keyframes (usually 300-700 keyframes); With the two buttons on the left side (previous/next buttons), the user can flip through the keyframes in the currently selected layer. The current temporal position in the programme is indicated with the timeline bar above the buttons. The layers have navigational links between them, meaning that when the user jumps up or down a layer, the current point of browsing is maintained. This browser is meant to be used with both hands holding the device and continuously tapping buttons in a highly interactive manner as if playing a pocket video game console.

Examples of the same sort of design approach applied to the Compaq iPAQ with its more detailed screen and 4,096 colours are shown in Figures 2 and 3. The different screen format dictates a different layout, but all the features met with the Revo are present: a small number of simple-to-use buttons, single or two-handed use, scroll-bars and level indicators, textual presentation used where necessary, but sparingly, and a clear visual presentation.

In the PDA interface designs, the user has full control over the displayed information on the screen with the widgets being very obvious and always in easy reach. This style makes it acceptable and natural for the user to casually take attention away from the screen and a few seconds later focus back on the screen. The interfaces were designed in such a way that the user need not pay careful visual attention or point at a small area in the middle of the screen, unlike the majority of desktop application interfaces. The Revo optionally provides "tick" sound as aural feedback whenever a screen is touched. However, mapping the "virtual buttons" on the above interfaces to physical buttons on the device would enhance the tactile feedback for the user.

5 CONCLUSION

In this paper, designing keyframe browsing interfaces for video in a desktop environment and a PDA environment is considered, with a specially constructed design framework as a base. The commercial and research community are more and more aware of the importance of recognising people's individual differences and personal preferences. In the user interface design field, attempts to cater for the diversity makes it difficult to have a single user interface for a system which supports everybody's needs. Furthermore, the diversification of different devices for very different environments makes it impossible to stick to a single interface to support these different environments. Identifying all possible interface elements and specifying an interface from this list can be a good starting step for heading toward realising universal access which supports potentially all users and their circumstances. This is the first step toward designs which automatically identify each individual user's preferences and needs at the time of use, and assemble suitable interface elements to provide this to the user dynamically.

Mere technological progress does not guarantee a wide acceptance of usage of that technology in the end product. Numerous failures in usability are found in small, handheld devices because the same interface paradigm for the so-far dominant desktop systems were used without further elaborate consideration. It is thus important to consider in depth the context of the use of the particular interface in concern.

References

[1] P. Browne, A. Smeaton, N. Murphy, N. O'Connor, S. Marlow and C. Berrut, "Evaluating and Combining Digital Video Shot Boundary Detection Algorithms", *Irish Machine Vision and Image Processing Conference (IMVIP 2000)*, QUB, Belfast, Northern Ireland, 31 August - 2 September 2000.

[2] B. Smyth, and P. Cotter, "A personalized television listings service", *Communications of the ACM*, 43(8), 107-111, 2000.

[3] H. Lee, A. Smeaton, P. McCann, N. Murphy, N. O'Connor and S. Marlow. "Físchlár on a PDA: A Handheld User Interface to a Video Indexing,

Browsing and Playback System", Poster presented in *6th ERCIM Workshop "User Interfaces for All."*, Florence, Italy, 25-26 October 2000.

[4] K. Mc Donald, A. Smeaton, S. Marlow, N. Murphy and N. O'Connor, "Online Television Library: Organisation and Content Browsing for General Users", *SPIE Electronic Imaging: Storage and Retrieval for Media Databases 2001*, San Jose, CA, 24-26 January 2001.

[5] N. O'Connor, S. Marlow, N. Murphy, A. Smeaton, P. Browne, S. Deasy, H. Lee, and K. Mc Donald, "Físchlár: An on-line system for indexing and browsing of broadcast television content", to appear in the *26th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, UT.

[6] http://lorca.compapp.dcu.ie/Video/

[7] A. MacLean, R. Young, and T. Moran, "Design rationale: the argument behind the artifact", *Proceedings of the ACM Conference on Wings for the Mind (CHI '89)*, Austin, TX, pp.247-252, 1989.

[8] C. Stary, "A structured contextual approach to design for all", *Proceedings of the 6th ERCIM Workshop "User Interfaces for All"*, Florence, Italy, 83-97, 2000.

[9] H. Lee, A. Smeaton, C. Berrut, N. Murphy, S. Marlow and N. O'Connor, "Implementation and analysis of Several Keyframe-based Browsing Interfaces to Digital Video", *Proc. 4th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2000*, Lisbon, Portugal, 18-20 Sept., 2000.

[10] H. Lee, *User Interface Design for keyframe-based Content Browsing of Digital Video*, PhD Thesis submitted to Dublin City University, January 2001.

[11] H. Zhang, C. Low, S. Smoliar and J. Wu, "Video parsing, retrieval and browsing: an integrated and content-based solution", *Proceedings of ACM International Conference on Multimedia* 9 5, San Francisco, CA, November 7-9, 1995, 503-512.

[12] "DVB-VCR", *SMASH(Storage for Multimedia Applications Systems in the Home) Project final report*, 1998.

[13] M. Mills, J. Cohen and Y-Y. Wong, "A Magnifier tool for video data", *Proceedings of ACM CHI '92*, 1992.

[14] S. Kristoffersen and F. Ljungberg, " 'Making place' to make IT work: empirical explorations of HCI for mobile CSCW", *Proceedings of the ACM Conference on Supporting Group Work (GROUP '99)* Phoenix, AZ, 276-285, 1999.

Figure 2: An example browsing interface on a Compaq iPAQ.

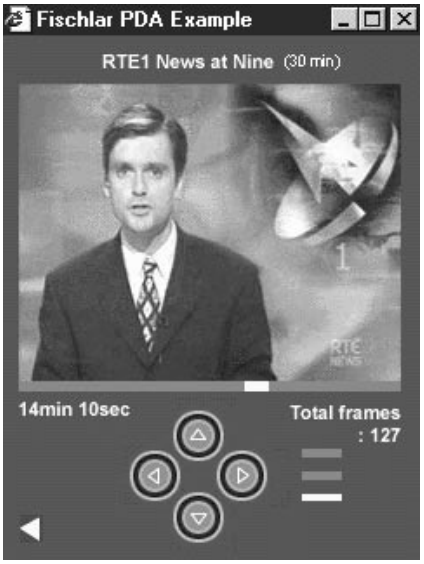


Figure 3: An example browsing interface on a Compaq iPAQ.

Figure 4: A Timeline Bar Browser screen shot with a playing video window superimposed.

Figure 5: The icons a Físchlár user can use to alter the video browsing format.

Figure 6: An example browsing interface on a Psion Revo.

Figure 7: An example browsing interface on a Psion Revo.

COLOR REFINEMENT FOR CONTENT-BASED IMAGE RETRIEVAL

Aamir Saeed Malik, Humaira Nisar, Tae-Sun Choi,
Kawangju Institute of Science and Technology
1 Oryong-dong, Puk-gu, Kwangju, 500-712, Korea

ABSTRACT

Color histograms, because of their efficiency and insensitivity to small changes, are widely used for content based image retrieval. But the main disadvantage of color histograms is that many images of different appearances can have similar histograms because color histograms provide coarse characterization of an image. In this paper, the technique defined is based on Histogram Refinement [1] and we call it Color Refinement. Color refinement splits the pixels in a given bucket into several classes just like histogram refinement method. The classes are all related to color and are based on color coherence vector.

1. INTRODUCTION

There are many techniques available for image retrieval. Roughly they can be classified into three categories, i.e., text-based retrieval, content-based retrieval and semantic-based retrieval. Each of the retrieval categories has their own strengths and weaknesses. This paper deals with the content-based retrieval.

There are queries that require the comparing of the images on their overall appearance. In such cases, color histograms can be employed because they are very efficient regarding computations. Plus they offer insensitivity to small changes regarding camera position. But the main problem with color histograms is their coarse characterization of an image. That may itself result in same histograms for images with different appearances. Color histograms are employed in systems such as QBIC [2], Chabot [3] etc.

In this paper, a modified scheme based on histogram refinement [1] method is presented. The histogram refinement method provides that the pixels within a given bucket be split into classes based upon some local property and these split histograms are then compared on bucket by bucket basis just like normal histogram matching but the pixels within a bucket with same local property are compared. So the results are better than the normal histogram matching. So not only the color feature of the image is used but also the spatial information is incorporated to refine the histogram.

First partition in each bin is based on spatial coherence of pixels just like computed by Pass and Zabih [1]. A pixel is coherent if it is a part of some sizable similar colored region, otherwise it is incoherent. Then two more properties are calculated for each of the coherent and incoherent pixels in each bin. First the number of clusters are found for each case, i.e., coherent and incoherent case. Secondly, the average of each cluster is computed.

The outline of this paper is as follows. Section two reviews the related work on image content retrieval based on color and spatial information. Section three describes the color refinement approach. Section four discusses the results obtained by testing the algorithm on a database of images provided in CD 6 and CD 8 of MPEG-7 data. Section five contains the conclusions. Finally references are given.

2. PREVIOUS WORKS

Research is being conducted by many around the world to incorporate the spatial information in addition to color in color histograms. Many methods and algorithms have been proposed.

Hsu [4] exploits the degree of overlap between regions of the same color. In their method, image is partitioned into rectangular regions after the selection of some representative colors from the image. Each of the rectangular regions has predominantly one color. The results of their experiments show that their method provides a better result than normal color histograms. They used a database of 260 images.

Smith & Chang' s method also partitions the image. However, they allow each region to contain multiple different colors instead of one predominant color like in Hsu method described above. Each pixel in their method may belong to several different regions. Histogram back-projection method [5] is used for back projecting set of colors onto the image. Finally, color sets with large connected components are identified. They used database of 3100 images for testing purposes.

Rickman and Stonham [6] provides a method based on small equilateral triangles with fixed sides. They randomly sample pixels at the endpoints of the equilateral triangles. Hence, triplets result for each triangle. They compare the distribution of these triplets. They used a database of 100 images.

Stricker and Dimai [7] finds the first three moments of the color distributions in an image. They use the HSV colorspace and they compute moments for each color channel. Their method is insensitive to small rotations and translations because they divide the image into partially overlapping regions. They used a database of about 11,000 images.

Huang et al. [8] method is called Color Correlogram and it captures the spatial correlation between colors. It is based on finding the probability that a pixel with color i will be k pixels away from a pixel of color j. They used a database of 18,000 images.

Pass and Zabih [1] method is Histogram Refinement. They partition histogram bins by the spatial coherence of pixels. They further refine it by using additional feature. The additional feature used is the center of the image. The center of the image is defined as the 75% centermost pixels. Their database consists of 14,554 images.

3. COLOR REFINEMENT

Color Refinement is proposed as a descriptor for MPEG-7. Color Refinement is based on histogram refinement [1] method. The histogram refinement method provides that the pixels within a given bucket be split into classes based upon some local property and these split histograms are then compared on bucket by bucket basis and the pixels within a bucket with same local property are compared.

Pre-Processing:

Three different methods can be used for preprocessing:

a) Convert to HSV Space. Quantize so as to obtain 8:2:2 (HSV) from 256:256:256 (RGB). Then obtain the histogram.
b) Convert to HSV Space. Quantize so as to obtain 8:2:2 (HSV) from 256:256:256 (RGB). Consider only the hue value. Then obtain the histogram.
c) Convert to grayscale intensity image. Uniformly quantize into eight quantized values. Then obtain the histogram.

Methods (b) and (c) are considered for preprocessing so as to reduce the feature vector size which is associated with the image. For a retrieval system to be successful, the feature vector f(I) should have certain desirable qualities: (i) $|f(I)-f'(I)|$ should be large if and only if I and I′ are not similar, (ii) $f(\bullet)$ should be fast to compute, and (iii) f(I) should be small in size.

Color Refinement Method:

Color histogram buckets are partitioned based on spatial coherence just like computed by Pass and Zabih [1]. A pixel is coherent if it is a part of some sizable similar colored region, otherwise it is incoherent. So the pixels are classified as coherent or incoherent within each color bucket. If a pixel is part of a large group of pixels of the same color which form at least one percent of the image then that pixel is a coherent pixel and that group is called the coherent group or cluster. Otherwise it is incoherent pixel and the group is incoherent group or cluster.

Then two more properties are calculated for each of the coherent and incoherent pixels in each bin. First the number of clusters are found for each case, i.e., coherent and incoherent case in each of the bin. Secondly, the average of each cluster is computed. So for each bin, there are six values: one each for percentage of coherent pixels, percentage of incoherent pixels, number of coherent clusters, number of incoherent clusters, average of coherent cluster and average of incoherent cluster.

These values are calculated by computing the connected components. A connected component C is a maximal set of pixels such that for any two pixels $p,p' \in C$, there is a path in C between p and p′. Eight connected neighbors method is used for computing connected component. A pixel is classified as coherent if it is part of a connected component whose size is equal to or greater than τ (τ = 1% of the image size). Otherwise it is classified as incoherent. And the connected component is classified as coherent connected component if it equals or exceeds τ. Otherwise it is classified as incoherent connected component.

Finally the average for coherent connected component is simply calculated since the number of coherent pixels and the number of coherent connected components are already known. Similarly, the average for incoherent connected component is also calculated from the number of incoherent pixels and the number of incoherent connected components.

For each discretized color j, let the number of coherent pixels as α_j, the number of coherent connected components as $C_{\alpha j}$ and the average of coherent connected component as $\mu_{\alpha j}$. Similarly, let the number of incoherent pixels as β_j, the number of incoherent connected components as $C_{\beta j}$ and the average of incoherent connected component as $\mu_{\beta j}$. For each discretized color j, the total number of pixels are $\alpha_j+\beta_j$ and the color histogram summarizes the image as $<\alpha_1+\beta_1,\ldots, \alpha_n+\beta_n>$.

Post-Processing:

We use the L_1 distance to compare two images I and I′. Using the L_1 distance, the jth bucket' s contribution to the distance between I and I′ is:

$$\Delta_1 = |(\alpha_j-\alpha'_j)| + |(\beta_j-\beta'_j)| \quad (1)$$
$$\Delta_2 = |(C_{\alpha j}-C'_{\alpha j})| + |(C_{\beta j}-C'_{\beta j})| \quad (2)$$
$$\Delta_3 = |(\mu_{\alpha j}-\mu'_{\alpha j})| + |(\mu_{\beta j}-\mu'_{\beta j})| \quad (3)$$

So we get a very finer distinction with this method. In original scheme [1], only equation (1) is used and in using only color histograms for comparison, the following equation is used:

$$\Delta_1 = |(\alpha_j+\beta_j)| - |(\alpha'_j+\beta'_j)| \quad (4)$$

Also equations (1) to (3) provide for incorporating the scalability. And remove problems identified by Huang et al. [8] which cannot be removed by only using CCV (Color Coherent Vector) defined in [1].

4. EXPERIMENTAL RESULTS

We implemented the color refinement and used it for image retrieval from a database of images provided in CD 6 and CD 8 of the MPEG-7 test material. We conducted the tests for methods (b) and (c) listed in the pre-processing stage. This was done to reduce the feature vector size.

We obtained six values for each of the bucket in the histogram. The six values include percentage of coherent pixels (α_j), percentage of incoherent pixels (β_j), number of coherent clusters ($C_{\alpha j}$), number of incoherent clusters ($C_{\beta j}$), average of coherent cluster ($\mu_{\alpha j}$) and average of incoherent cluster ($\mu_{\beta j}$) for each jth bucket. We used total of eight buckets. So the total length of the feature vector associated with an image is 48 integer values.

We compared the results with L_1 distance. First, we used equation (1) for identifying the similarity between images. Then we used equation (2) to further refine the results and finally we used equation (3) to get the final result.

Images in CD 6 and CD 8 are in PCD format. They were converted to JPEG format for simulations. Descriptor values for some of the still images from CD 6 and CD 8 are provided in Appendix 1 and Appendix 2. As can be seen from the appendices, the length of the

feature vector can be further reduced in case if any of α_j or/and β_j is zero.

5. CONCLUSION

Usage of method (a), described in pre-processing, give better results. However, the length of the feature vector is increased from 48 integer values to 192 integer values. So, that method was not implemented.

If color refinement is used as a descriptor for MPEG-7 than it takes care of the color as well as the spatial relation feature. And hence, it provides better results than the equivalent methods.

However, computational complexity is increased. But since the speed of the retrieval program is not the criteria for evaluation so the computational complexity and its effect on the speed of the retrieval program will be considered in next stage.

Appendix 1: Descriptor Values for Images from CD 6

Bin 1

Image #	α_j	β_j	$C_{\alpha j}$	$C_{\beta j}$	$\mu_{\alpha j}$	$\mu_{\beta j}$
Img0017	0	0	0	0	0	0
Img0018	0	0	0	0	0	0
Img0019	0	0	0	0	0	0
Img0041	0	0	0	0	0	0
Img0042	0	0	0	0	0	0
Img0043	0	0	0	0	0	0
Img0085	0	0	0	0	0	0

Bin 2

Image #	α_j	β_j	$C_{\alpha j}$	$C_{\beta j}$	$\mu_{\alpha j}$	$\mu_{\beta j}$
Img0017	98	2	1	11	8178	16
Img0018	100	0	1	5	14503	1
Img0019	99	1	1	5	14897	25
Img0041	98	2	2	1	7596	233
Img0042	100	0	1	23	11772	1
Img0043	98	2	2	20	8136	17
Img0085	98	2	1	8	2283	5

Bin 3

Image #	α_j	β_j	$C_{\alpha j}$	$C_{\beta j}$	$\mu_{\alpha j}$	$\mu_{\beta j}$
Img0017	0	100	0	19	0	8
Img0018	0	100	0	18	0	7
Img0019	0	100	0	8	0	11
Img0041	0	100	0	2	0	7
Img0042	0	100	0	4	0	6
Img0043	0	100	0	7	0	9
Img0085	0	100	0	14	0	16

Bin 4

Image #	α_j	β_j	$C_{\alpha j}$	$C_{\beta j}$	$\mu_{\alpha j}$	$\mu_{\beta j}$
Img0017	0	100	0	6	0	4
Img0018	0	100	0	9	0	5
Img0019	0	100	0	4	0	3
Img0041	0	0	0	0	0	0
Img0042	0	0	0	0	0	0
Img0043	0	0	0	0	0	0
Img0085	0	100	0	16	0	8

Appendix 2: Descriptor Values for Images from CD 8

Bin 5

Image #	α_j	β_j	$C_{\alpha j}$	$C_{\beta j}$	$\mu_{\alpha j}$	$\mu_{\beta j}$
Img0001	0	0	0	0	0	0
Img0002	0	0	0	0	0	0
Img0020	0	100	0	26	0	14
Img0061	0	100	0	18	0	4
Img0062	0	100	0	19	0	6

Bin 6

Image #	α_j	β_j	$C_{\alpha j}$	$C_{\beta j}$	$\mu_{\alpha j}$	$\mu_{\beta j}$
Img0001	0	0	0	0	0	0
Img0002	0	0	0	0	0	0
Img0020	0	100	0	17	0	28
Img0061	89	11	2	22	1353	15
Img0062	39	61	1	46	510	17

Bin 7

Image #	α_j	β_j	$C_{\alpha j}$	$C_{\beta j}$	$\mu_{\alpha j}$	$\mu_{\beta j}$
Img0001	89	11	4	15	804	26
Img0002	85	15	3	16	1051	35
Img0020	62	38	2	23	413	22
Img0061	17	83	1	53	257	23
Img0062	66	34	2	29	893	32

Bin 8

Image #	α_j	β_j	$C_{\alpha j}$	$C_{\beta j}$	$\mu_{\alpha j}$	$\mu_{\beta j}$
Img0001	99	1	3	2	3229	26
Img0002	99	1	3	16	3580	7
Img0020	87	13	3	37	1721	20
Img0061	61	39	3	60	894	28
Img0062	91	9	2	42	3529	17

REFERENCES

[1] Greg Pass and Ramin Zabih. Histogram Refinement for content−based image retrieval. In IEEE Workshop on Applications of Computer Vision, pages 96-102, December 1996.

[2] M. Flickner et al. Query by image and video content: The QBIC system. IEEE computer, 28(9):23-32, September 1995.

[3] Virginia Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. IEEE computer, 28(9):40-48, September 1995.

[4] Wynne Hsu, T. S. Chua and H. K. Pung. An integrated color-spatial approach to content based image retrieval. In ACM Multimedia Conference, pages 305-313, 1995.

[5] Michael Swain and Dana Ballard. Color indexing. International Journal of Computer Vision, 7(1):11-32, 1991.

[6] Rick Rickman and John Stonham. Content based image retrieval using color tuple histograms. SPIE proceedings, 2670:2-7, February 1996.

[7] Markus Stricker and Alexander Dima. Color indexing with weak spatial constraints. SPIE proceedings, 2670:29-40, February 1996.

[8] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In IEEE Conference on Computer Vision and Pattern Recognition, pages 762-768, 1997.