CHANGE DETECTION-BASED VIDEO SEGMENTATION FOR SURVEILLANCE APPLICATIONS

Paulo Lobato Correia, Fernando Pereira

Instituto Superior Técnico – Instituto de Telecomunicações, 1049-001 Lisboa, Portugal
e-mail: {Paulo.Correia, Fernando.Pereira}@lx.it.pt

ABSTRACT

This paper proposes an automatic video segmentation algorithm for indoor surveillance applications. The change detection segmentation algorithm is used to locate intruders into a sensitive area being monitored by a video camera. For this type of application, the relevant objects to segment are typically unknown, preventing the inclusion of extensive *a priori* information; however, the precision of object contours is not extremely important, as long as the intruders are effectively detected. Once detected, the video information can be selectively encoded, e.g., using the MPEG-4 standard, providing a good image quality for the intruder object, while guaranteeing a low overall bit-rate. This enables video transmission of the intrusion detection over a low speed communications network.

1. INTRODUCTION

An indoor surveillance application typically uses a video camera to monitor an area inside a building with the purpose of automatically detecting intruders. One architecture of the system includes a video camera, monitoring the selected location, with a communications connection to a central security office.

When an intruder is detected, the surveillance system deploys an alarm condition and may perform different types of actions depending on the configuration and capabilities of the system. The simplest action may be to activate a video recording device while notifying the security responsible, and sounding an alarm.

Since one central security office may provide surveillance services to a number of geographically distributed sites, e.g., within a city, the video images corresponding to the detected alarm conditions may have to be transmitted to this central security office using a telecommunications connection. If low bandwidth connections are to be used (e.g., a fixed or mobile telephone line), then the video content representation should be as efficient as possible. For this purpose an MPEG-4 encoding [1] solution may be employed, using two objects: the intruder encoded with the best possible quality, and the remaining image encoded with a lower quality. Eventually, only the intruder object may be transmitted. The application thus requires a segmentation of the scene, to identify the intruder and background objects.

A variation of this application may be designed to check whether a detected person is an employee of the company being monitored, and if that person has authorization for accessing the building at that time. In this case, the surveillance application should also include a face detection and recognition algorithm to be applied to the image area corresponding to the intruder. Or, the video camera can be automatically adjusted (zoom, pan, tilt) to give a more detailed image of the intruder, in response to the information provided by the segmentation algorithm.

The proposed surveillance segmentation solution has been tested using the sequences: *Hall Monitor* (a camera is used to monitor a corridor within an office environment) [2] and *Stair Wide* (a camera monitoring the entrance of a building) [3] – see sample images in Figure 1 (a) and (b), respectively.

(a) (b)

Figure 1 – Samples from the test sequences *Hall Monitor* (a), and *Stair Wide* (b).

2. PROPOSED SEGMENTATION SOLUTION

For the surveillance application considered, video cameras capture images of a static scene, with illumination changes, most of the time. The entrance of an intruder into the scene can thus be detected by the changes it causes.

A change detection segmentation algorithm can be used, with the changing areas typically corresponding to intruders. The algorithm must be able to deal with illumination changes, and should allow a fast implementation (appropriate for real-time operation).

The change detection algorithm proposed in this paper builds on ideas presented in [4, 5, 6, 7, 8]. It implements a statistical hypothesis test to decide whether a given pixel has changed or not, like in [6], and, additionally, the thresholding step makes extra considerations about the differences between the changed and unchanged areas' variances, and on the size of the changed area, to achieve a better behavior for the thresholding operation (which is determinant for the algorithm performance).

The concept of a change detection memory is used to increase the stability of the resulting segmentation, like in [7], but the control of the memory update is specific of this algorithm proposal.

The main modules of the proposed change detection segmentation, shown in Figure 2, are:

- *Thresholding* – Classification of pixels as changed or not results from the thresholding of the difference between consecutive images. The threshold value is automatically computed, according to the video sequence characteristics, without any manual configuration.

- *Combination with memory* – The thresholding output is combined with the segmentation masks from previous time instants, available from a memory, to make the change detection results more stable. This improves segmentation results when the motion of (parts of) a given object temporarily stops.
- *Smoothing* – Isolated pixels are removed and small holes in objects are filled to make the change detection segmentation result smoother.
- *Memory update* – The final step consists in the automatic adjustment of the memory contents, according to the observed sequence characteristics.

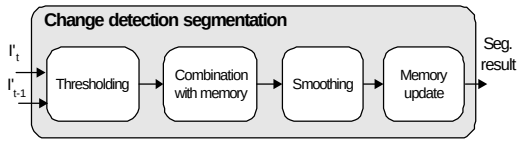


Figure 2 – Block diagram for the proposed change detection segmentation algorithm.

Brief descriptions of the thresholding and memory update steps, the major contributions of the proposed algorithm, are included in the following subsections.

2.1. Thresholding

The thresholding decision is made based on a hypothesis test. Inside the unchanged area, the variation of a pixel's luminance is assumed to be due to camera noise (n_c), and thus to have twice its variance (since a difference is involved). Assuming the camera noise has a normal distribution with zero mean and s_c^2 variance [6], the unchanged pixel differences (d_i) also have a normal distribution:

$$n_c \sim N\left(0, s_c^2\right) \qquad d_i \sim N\left(0, s_{back}^2\right)$$

With: $s_{back}^2 = 2 \cdot s_c^2$

The decision of whether a pixel has changed or not, is not completely uncorrelated from its neighbors. As such, the algorithm looks into an n x n window, centered on that pixel, and the sum of the squared difference values (S) within the window is considered to provide the desired context. This new variable S follows a chi-square distribution with n^2 degrees of freedom [9]:

$$S = \frac{1}{2 \cdot s_c^2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} d_i^2 = \frac{1}{2 \cdot s_c^2} \cdot \sum_{k=1}^{n^2} d_i^2 \sim c_{n^2}^2$$

Now, the probability of an unchanged pixel being erroneously classified as changed is given by:

$$1 - a = P\left(S > t_a \mid \text{pixel unchanged}\right)$$

Setting the probability of error (1- a) sufficiently low, say to 10^{-3}, a t_a value of 52.6 is found from the chi-squared distribution tables [10].

With this approach, the change detection thresholding can be performed according to:

$$\begin{cases} S > t_a & \rightarrow pixel\ changed \\ S \le t_a & \rightarrow pixel\ unchanged \end{cases}$$

To improve computation efficiency, a simplification has been adopted [6], assuming that for small windows small variations of pixel behavior are expected. Thus, S can be computed as:

$$S = \frac{25 \cdot d_i^2}{2 \cdot s_c^2}$$

The comparison can be performed between each pixel's squared difference value (d_i^2) and a threshold (Thr):

$$\begin{cases} d_i^2 > Thr & \rightarrow pixel\ changed \\ d_i^2 \le Thr & \rightarrow pixel\ unchanged \end{cases}$$

With: $Thr = \frac{t_a \cdot 2 \cdot s_c^2}{25} = 2.1 \cdot s_{back}^2$

To make the thresholding operation more dependent on the type of content being segmented, the value of the luminance difference variance in the images' changed areas (s_{fore}^2) has also been taken into account. The adaptive threshold value to be used is given by:

$$Thr = a \cdot s_{back}^2 + b$$

With:

$$a = \min\left\{5, \left\lfloor 2.1 + \max\left(0, \frac{\frac{s_{fore}^2}{s_{back}^2} - 15}{25}\right)\right\rfloor\right\} \qquad b = \max\left(0, \frac{15 - \frac{s_{fore}^2}{s_{back}^2}}{20}\right)$$

The modifications introduced have two main purposes: *(i)* when the two types of regions (changed and unchanged) have very different variances, the threshold is increased to guarantee a correct separation between them (factor a); *(ii)* when the two variances are very close, some unchanged areas may be erroneously classified as changed, and the increment b is added to improve the separation between the two types of regions.

Additionally, when the background variance is very low ($s_{back}^2 < 0.5$), to prevent using a threshold value too low, with the risk of erroneously classifying every noise peak as part of the changed area, the threshold used is:

$$Thr = 2.1 \cdot s_{back}^2 + 0.5$$

Finally, the threshold value is averaged with that of the previous time instant to ensure a smooth evolution.

2.2. Memory Update

The memory stores information about the changed areas detected in past time instants, being essential to keep track of objects even when they temporarily stop moving, to ensure a better temporal continuity of changed regions.

However, using a long memory may have the undesired effect of creating segmentation masks for the moving objects that are much larger than the actual objects.

The algorithm memory length control parameter (*MemLength*) represents the number time instants in which the pixel's classification as changed should be kept. This parameter is automatically adjusted according to the sequence characteristics, converging to zero when a considerable amount of motion is detected, and to the maximum allowed value when only slower motions are detected.

The initial and maximum *MemLength* values are set to 5 and 15. The automatic adjustments depend on:

- *NumChanged* – Number of changed pixels detected in the thresholding step for the current time instant.
- *PrevNumChanged* – Like *NumChanged*, but for the previous time instant.
- *TotalChanged* – Number of changed pixels detected after the combination with memory and smoothing steps for the current time instant.
- *PrevTotalChanged* – Like *TotalChanged*, for the previous time instant.
- *NewInNumChanged* – Number of changed pixels detected in the thresholding step for the current time instant, that were not detected as changed in the previous thresholding step.
- *NewInTotalChanged* – Number of changed pixels detected in the thresholding step for the current time instant, that were not detected as changed after the combination with memory and smoothing steps for the previous time instant.

The memory length is decreased by one whenever the object size is stable but motion is detected:

$$\left(\left|TotalChanged - PrevTotalChanged\right| < \frac{PrevTotalChanged}{10}\right) \wedge \left(\frac{1}{20} < \frac{NewInTotalChanged}{NumChanged} < \frac{1}{5}\right)$$

The memory length is increased by one when:

- The image content is very stable:

$$NewInTotalChanged \leq 0.03 \cdot TotalChanged$$

- Movement is due to new object parts moving, leading to an increase of the changed area size:

$$\left(\left|NumChanged - PrevNumChanged\right| > \frac{PrevNumChanged}{5}\right) \vee \left(NewInNumChanged > \frac{NumChanged}{3}\right) \vee \left(\left|TotalChanged - PrevTotalChanged\right| > \frac{PrevTotalChanged}{4}\right)$$

If no changed pixels are detected by the thresholding step, the initial *MemLength* value is adopted.

2.3. Other Considerations

To reduce the complexity and bit-rate of the subsequent encoding of the segmented images for their transmission to the surveillance control center, the shape of the objects identified by the change detection segmentation algorithm can be simplified by a post-processing operation.

In view of MPEG-4 encoding, producing rectangular object masks (containing the detected intruder objects) is considered. This can be applied if a precise segmentation of the intruder is not needed for the use the application.

3. RESULTS

The results presented use the QCIF format, as this is adequate for transmission when bandwidth restrictions are critical, notably for fixed or mobile telephone channels. However, the spatial format of the images does not significantly affect the performance of the change detection algorithm [11], and similar results are obtained for other image formats, e.g., CIF. Results for the Hall Monitor sequence are included in Figure 3, and also in Figure 4, after post-processing the change detection output to produce rectangular object masks. The intruder object is presented on the left and the background on the right.

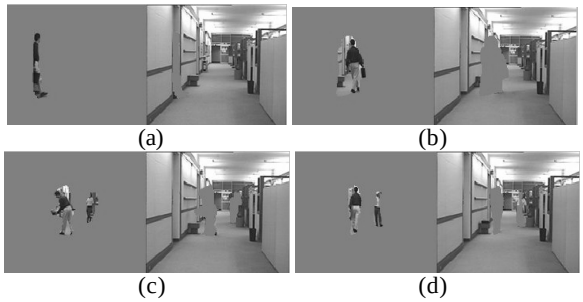


Figure 3 – Segmentation results for images number 20 (a), 60 (b), 100 (c) and 160 (d) of the Hall Monitor sequence.

Looking at the full set of results obtained with the change detection algorithm for the Hall Monitor sequence, it is possible to conclude that the segmentation results are appropriate for the surveillance application considered, as the intruders are correctly detected.

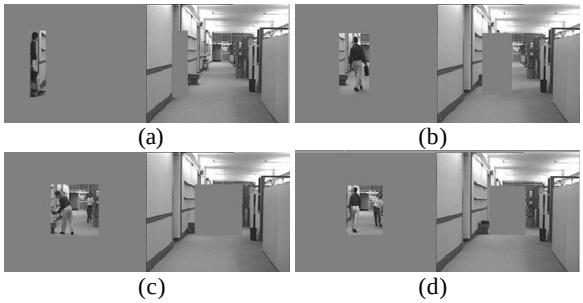


Figure 4 – Seg. results for images 20 (a), 60 (b), 100 (c), 160 (d) of Hall Monitor sequence, using post-processing.

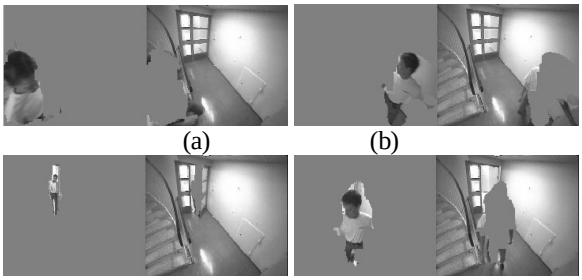
The results using the post-processing operation can also be used for the surveillance application, as the larger intruder area obtained does not seem to be a disadvantage for human inspection. And, using simpler object shapes has advantages for MPEG-4 encoding – see Table 1.

Table 1 – Average PSNR and percentage of bits used for encoding the shape using an MPEG-4 Visual Core profile encoder, for the *Hall Monitor* sequence, using the change detection algorithm with and without post-processing, with a target bitrate of 64 kbit/s.

	Object	Y PSNR (dB)	U PSNR (dB)	V PSNR (dB)	Shape bits (%)
CD	Back.	33.8	37.9	40.1	28.3
	Men	26.3	33.2	36.0	15.2
CD + pos.proc	Back.	35.7	38.7	40.8	12.5
	Men	27.9	34.1	36.5	2.4

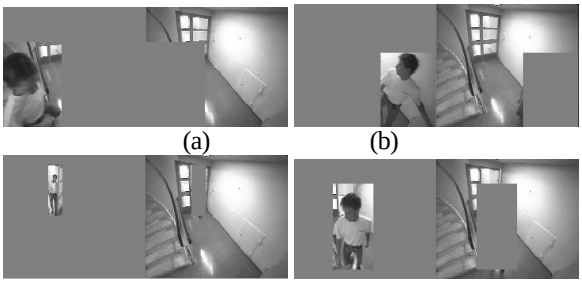
The results in Table 1 were obtained using the MPEG-4 Visual Core profile [8], with a total target bit-rate of 64 kbit/s (20 kbit/s for the background and 44 kbit/s for the walking men). The results mainly reflect the different texture/shape allocation of each object's target bit-rate caused by the different complexity of the shapes to be encoded. As expected, simpler shapes require fewer coding resources, leaving a larger portion of the bit-rate for texture encoding, leading to higher PSNR values.

Results obtained for the Stair Wide sequence (25 Hz) are included in Figure 5 and Figure 6, using the proposed change detection algorithm, without and with the post-processing, respectively.



(a) (b) (c) (d)

Figure 5 – Segmentation results for images 940 (a), 1000 (b), 1300 (c) and 1350 (d) of *Stair Wide* sequence using the proposed change detection algorithm.



(a) (b) (c) (d)

Figure 6 – Segmentation results for images 940 (a), 1000 (b), 1300 (c) and 1350 (d) of *Stair Wide* sequence using the change detection algorithm with post-processing.

As for the previous sequence, the adaptive thresholding of proposed change detection algorithm enables a correct detection of the intruders.

4. CONCLUSIONS

The goal of the indoor surveillance application discussed in this paper is to detect intruders entering a sensitive area of a building with a very low probability of missing them. The segmentation solution proposed consists in a change detection algorithm, which does not require a complex implementation, and is effective in detecting the intruders. The proposed algorithm includes a novel adaptive threshold selection procedure, as well as a contribution in the memory update procedure.

In this type of scenario, the contour precision of the objects may not be critical, as long as the intruder object is effectively detected. In this context, a post-processing operation has been considered to produce object masks with simple (rectangular) shapes to improve the coding efficiency, in view of their transmission to a central surveillance office whenever an alarm condition occurs. Notice that even when adopting the rectangular object mask solution, functionalities such as face detection for posterior recognition, are made easier as the search area is effectively reduced.

5. ACKNOWLEDGMENT

We wish to acknowledge the support provided by the European Network of Excellence VISNET (IST Contract 506946).

6. REFERENCES

[1] ISO/IEC 14496, *Information Technology – Coding of Audio-Visual Objects*

[2] MPEG Video Group, *MPEG-4 Video Verification Model 16.0*, Doc. ISO/IEC JTC1/SC29/WG11 N3312, Noordwijkerhout MPEG meeting, March 2000

[3] *Mobile Multimedia Systems*, ACTS project - AC098, www.tnt.uni-hannover.de/project/eu/momusys/overview.html

[4] M. Hötter and R. Thoma, "Image Segmentation Based on Object Oriented Mapping Parameter Estimation", *Signal Processing*, Vol.15, pp. 315-348, 1988

[5] R. Thoma and M. Bierling; "Motion Compensating Interpolation Considering Covered and Uncovered Background", *Signal Processing: Image Communication*, Vol. 1, pp. 191-212, 1989

[6] T. Aach and A. Kaup, "Statistical Model-Based Change Detection in Moving Video", *Signal Processing*, 31, pp. 165-180, 1993

[7] Roland Mech and Michael Wollborn; "A Noise Robust Method for 2D Shape Estimation of Moving Objects in Video Sequences Considering a Moving Camera", *Signal Processing* - Special Issue on "Video Sequence Segmentation for Content-Based Processing and Manipulation", Vol. 66, No. 2, pp. 203-217, 1998

[8] ISO/IEC 14496-2:2001, *Information Technology – Coding of audio-visual objects – Part 2: Visual*, 2001

[9] P. Meyer, *Probabilidade – Aplicações à Estatística*, Livros Técnicos e Científicos Editora, 2nd Edition, 1983

[10] C. Croarkin and P. Tobias, (Technical Editors), *Engineering Statistics Internet Handbook*, NIST/SEMATECH, available at www.itl.nist.gov/div898/handbook/index.htm

[11] Paulo Correia, *Video Analysis for Object-Based Coding and Description*, PhD Dissertation, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal, 2002