KEY-FRAME RADIAL PROJECTION FOR ROBUST VIDEO HASHING

C. De Roover, F. Lefebvre, C. De Vleeschouwer and B. Macq
Telecommunications and Remote Sensing Laboratory, UCL, Belgium

ABSTRACT

Robust signal hashing defines a feature vector that characterizes the signal, independently of non-significant distortions of its content, due for example to compression. Our paper first proposes a robust image hashing algorithm, based on radial projection of the image pixels, and evaluates its robustness and discriminating capabilities. The method appears to generate similar feature vectors for visually equivalent images, while resulting in distinct vectors for two different images. Robust image hashing is then extended to video sequences by selecting key-frames, and by extracting a feature vector for each of these frames. Experiments demonstrate that the proposed approach is able to characterize the video sequence content, while providing good robustness towards spatial or temporal distortions.

1. INTRODUCTION

Accessing, organizing, and managing visual contents present technical challenges due to the large and always growing amount of available content, and to the lack of normalized and reliable ways to describe the image attributes. Following the terminology introduced in recent literature about visual content authentication [1, 2, 3, 4], we use the term "hashing" to denote the extraction of an image-based feature vector, but make the distinction between cryptographic and robust hashing. A cryptographic hash, which is generally used for digital signature, summarizes and uniquely identifies a message by a short and constant bit length feature vector. In cryptography, the feature vector produced by a hash function dramatically changes when a single bit of the input message changes. One says that cryptographic feature vectors are discontinuous. On the contrary, a continuous or robust hash function alters the feature vector in proportion to the changes in the input signal.

Based on the above definitions, in the context of image processing, robust hashing has to generate similar feature vectors for visually equivalent images, while resulting in distinct vectors for two different images. As a consequence, the comparison of the image feature vectors computed by a robust hashing algorithm can indicate whether the corresponding images are visually similar or not, independently of non-significant distortions due to common manipulations like compression or re-sampling. Because it defines a vector that characterizes the image content, robust hashing is an obvious solution for content identification and indexing. When used in combination with conventional cryptographic digital signature methods, robust hashing can also support robust integrity and authentication systems [1, 5, 6, 7].

Our paper proposes a robust image and video hashing algorithm. Section 2 and 3 describe how to compute the image and video feature vectors. Section 4 validates our methodology.

Work funded by the Region Wallonne and by the Belgian NFS.

2. ROBUST IMAGE HASHING BASED ON RADIAL VARIANCE PROJECTIONS

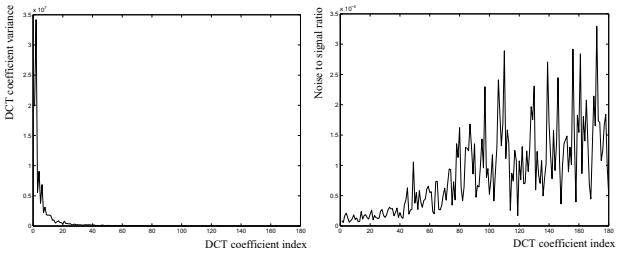
Our proposed robust hashing algorithm works in two steps. First, a set of radial projections of the image pixels compute the RAdial Variance (RAV) vector. Second, the discrete cosine transform of the RAV vector defines the Transformed RAV (TRAV) vector, whose first 40 coefficients form our proposed robust image feature vector, denoted Radial hASH (RASH) vector. We now motivate our approach, and explain how the radial projections are computed.

The components of the RAV vector are computed on a set of lines articulated around the center of the image. Defining the feature vector based on radial projections provides resilience to scaling and rotation around the center of the image. In practice, we use a 180-sample feature vector, which corresponds to a uniformly distributed set of 180 angles ϕ, with $0 \leq \phi < 180$. For each projection angle, the RAV sample is defined as the variance (and not the mean as originally proposed by the authors in [8]), of the pixels luminance values along the corresponding line. The intuition that lies behind the choice of the variance is the following. As a second order moment, the variance efficiently captures luminance discontinuities along the line. In the image, these discontinuities correspond to edges that are orthogonal to the projection direction. As a result, the variance is expected to capture relevant information about the distribution of the edges in the image, which in turns characterizes the visual content of an image.

The Discrete Cosine Transform (DCT) is used to extract a compact image feature vector from the redundant RAV vector. Figure 1 motivates the DCT coefficients selection. In Figure 1(a), the variance of the RAV DCT coefficients shows that most of the RAV vector energy is compacted on few low-frequency DCT coefficients. We conclude that the DCT efficiently decorrelates the RAV samples. In addition, Figure 1(b) shows that the noise-to-signal ratio due to JPEG compression remains almost constant for the 40 first low-frequency RAV DCT coefficients, but progressively increases for higher order coefficients. Based on these observations, we define the RASH image feature vector as the 40 low frequency coefficients of the DCT transformed RAV feature vector.

3. VIDEO HASHING BASED ON REPRESENTATIVE KEY-FRAMES

A naive way to extend "image hash" to "video hash" is to compute an image feature vector for each frame of the video sequence. However, this approach is computationally expensive, results in high dimensional video feature vector, and is significantly affected by temporal re-sampling of the video sequence. To circumvent these drawbacks, we notice that most real-life video sequences can be temporally divided into video shots [9], defined as groups of successive frames that are visually similar. Therefore, we decide to describe the video sequence as a set of feature vectors, one vector



(a) Expected energy. (b) Noise-to-signal ratio.

Fig. 1. RAV vector DCT statistics as a function of the coefficient index. Statistics are based on a 40-images subset of the USC-SIPI database. Noise-to-signal ratio is derived from the RASH error resulting from JPEG compression of the input images.

being associated to each video shot. In our case, each feature vector is the RASH vector of a carefully selected *key-frame*. To select the key-frames, we first identify some easily detectable frames, the video-shot boundary frames, simply denoted boundary-frames in the following. Once the boundary-frames have been located, one key-frame is selected between each pair of consecutive boundary-frames. We now formally define boundary- and key- frames.

Video-shot boundary-frames separate groups of (visually) similar frames. They can be detected by large disparity measurements between two successive frames. A common feature to evaluate the disparity between video frames is the luminance histogram, and the ℓ_1 norm is recommended to compute histogram differences [10, 11]. So, letting $H_k(j)$ denote the j^{th} component of the 64-bins luminance histogram of the k^{th} frame, the disparity $d(k, k-1)$ measured between frame k and $(k-1)$ is:

$$d(k, k-1) = \sum_{j=1}^{64} |H_k(j) - H_{k-1}(j)|, \quad (1)$$

Once the distance $d(k, k-1)$ has been computed, it is compared to a threshold to decide whether frame k is a boundary-frame or not. Next to heuristic thresholds, automatic thresholds, either global or adaptive, have been proposed in the literature [12, 13]. We propose to combine both kinds of thresholds as follows.

As in [12], a (pseudo-)global threshold, denoted $T_{global}(k)$, is defined on a large sliding window of size $2L+1$, centered on frame k. $\mu(k)$ and $\sigma(k)$ denoting the mean and the standard deviation of the disparity values, $d(i, i-1)$, measured with i in $[k-L, k+L]$, we define

$$T_{global}(k) = \mu(k) + \alpha_1 \sigma(k). \quad (2)$$

As in [13], a local and adaptive threshold, denoted $T_{local}(k)$, is computed on a small sliding window of size $2S+1$, with $S << L$, centered on frame k. We have

$$T_{local}(k) = \alpha_2 d_{max2}(k) \quad (3)$$

where $d_{max2}(k)$ is the second maximum value of $d(i, i-1)$, measured for $i \in [k-S, k+S]$.

Based on (2) and (3), frame k is defined to be a boundary-frame if $d(k, k-1)$ is the maximum disparity measured on the window of size $2S+1$ and centered in k, and if $d(k, k-1) > max(T_{global}(k), T_{local}(k))$. In our experiments, we use : $S = 10$, $L = 50$, $\alpha_1 = 3$, and $\alpha_2 = 2$.

The experiments presented in Section 4.2.1 demonstrate that both thresholds complement each other. In short, we can say that the global threshold avoids detecting non-relevant boundary-frames in periods of weak disparities, while the adaptive threshold prevents the detection of too many boundary-frames in periods of high disparities.

Once the shots have been identified, a key-frame is selected between each pair of consecutive boundary-frames, to characterize the corresponding video-shot visual content. Key-frames detection has been extensively discussed in previous works [14, 15]. For simplicity, we have chosen a simple approach. Given the indices k_1 and k_2 of two consecutive boundary-frames, the index r of the key-frame is defined by $r = \arg\min_{k_1 < k \leq k_2} d_{l_1}(k, k-1)$. Representativeness of selected frames is discussed in Section 4.2.2.

4. EXPERIMENTAL VALIDATION

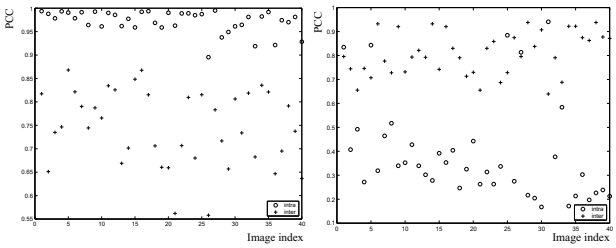
4.1. Image hash experimental validation

As explained in the introduction, a robust image hashing algorithm has to generate (i) equivalent outputs for visually similar inputs (= robustness), and (ii) different outputs for visually distinct input images (= collision avoidance). In this section, we evaluate the robustness and collision avoidance capabilities of our proposed hashing method. For comparison purpose, we also provide results based on the 64-bin luminance histogram, which is a common image feature vector used in content-based retrieval system [9].

For each of the 40 images of the dataset, we consider 8 manipulations, generating 320 images, named *processed images* in the following. The 8 manipulations envisioned in the experiments are (i) 3x3 average and Gaussian filtering, (ii) JPEG compression with 80 and 60% quality factor, and (iii) scaling (factors = 1.2, 0.8), rotation (2^0), and rotation (1^0) followed by "inside-box" cropping.

In Figure 2, for each original image, we compare Intra and Inter matching. The matching between two frames is computed as the peak of cross-correlation (PCCs) between the feature vectors extracted from the two frames. Intra and Inter matching are defined with regards to a specific original image. Given an original image, the 320 processed images are split into Intra and Inter images, depending on whether they have been derived from the original image or not. Based on this classification, Intra (Inter) matching denotes the matching between the original image and an Intra (Inter) processed images. A good match corresponds to a high PCC value. As an example, in Figure 2, the worst Intra matching refers to the smallest PCC computed betwen feature vectors of an original image and any of the images derived from this original.

Figure 2(a) considers the RASH feature vector. We observe that all Intra PCC's are larger than 0.85, and that no Inter PCC lies above 0.85. We conclude that cross correlation is an efficient way to compare two RASH feature vectors, and that 0.85 is a good threshold to decide whether two images are visually similar or not. For comparison purposes, Figure 2(b) provides Intra and Inter matching based on the 64-bin histogram feature vector. We observe that in nearly all cases, the worst Intra matching is lower than the best Inter matching, which indicates that after processing, it is not possible to partition Intra and Inter images based on histogram comparisons, while it is the case with the RASH vector.



(a) RASH (b) Histogram

Fig. 2. Comparison of the worst Intra matching with the best Inter matching for each one of the 40 tested images taken from the USC-SIPI database.

4.2. Video hash experimental validation

This section presents a number of experimental results to demonstrate that the RASH vectors extracted from the key-frames of a video sequence are good candidates to characterize the video sequence, independently of the manipulations the sequence has undergone (compression, sub-sampling, small geometrical distortions). First, we study the robustness of the shot boundary detection algorithm, i.e. we analyze whether the same video shots are identified before and after video sequence processing. Doing so, we show that the shot boundary detection algorithm based on the combination of an adaptive and a (pseudo-)global threshold outperforms other approaches. Then, we evaluate the relevance in terms of sequence representativeness of the key-frames. We show that the combination of adaptive and global thresholds results in increased characterization of the video sequence. Finally, we validate the whole system by matching the key-frames that are selected in a processed video sequence, with the key-frames that are selected either from the corresponding original sequence, or from other video sequences. This computation demonstrates that, even for strong degradation of the processed sequence (PSNR lower than 25 dB), our method remains able to associate each processed video to its original version.

In this section, we consider three original sequences, each sequence being extracted from a DVD support. The sequences are Monster (1341 frames of 576x304 pixels), Swordfish (1364 frames of 642x272), and Star wars Episode I (1092 frames of 688x320 pixels). For each sequence, a processed video sequence is obtained by capturing with a camera the sequence displayed on a screen. The average PSNRs of all processed sequences lie between 23 and 25 dB, which corresponds to severe distortions. For each original and processed video sequence, we detect key-frames and compute their RASH vectors. These operations are performed in real time on a Pentium III, 500MHz, 512Ram.

4.2.1. Shot boundary-frames selection robustness

To validate the performance of a boundary-frame detection algorithm, we compare the set of boundary-frames detected in a given original sequence with the one detected in a corresponding processed sequence. We introduce the *equivalence* measurement, denoted θ, to quantify the similarity between the sets of original and processed boundary-frames. θ, which is associated to a given detection method, and to a pair of original and processed video sequences, is defined as

$$\theta = \frac{\#Correct}{\#Correct + \#False\,alarm + \#Missed} \quad (4)$$

In equation (4), "$\#Correct$" denotes the number of boundary-frames that are detected in both the original and the processed video sequences. The "Missed" boundary-frames are the frames that are detected in the original sequence but not in the processed sequence. The "False Alarm" frames denote the frames that are detected in the processed sequence, but not in the original sequence.

We now exploit θ to compare three shot boundary detection algorithms, using respectively the (pseudo-)global threshold defined by (2), the locally adaptive threshold defined by (3), or a combination of these two thresholds. Table 1 presents equivalence measurements for the 3 pairs of original and processed video sequences described above, and corresponding to Starwars I (SW1), Monster (Mon), and Swordfish (Swo). From Table 1, we conclude that combining (2) and (3) performs better than other approaches.

Threshold	SW1	Mon	Swo
Local	$80,9\%$	$64,3\%$	$73,9\%$
Global ($L = 20$)	$53,3\%$	$32,2\%$	$67,8\%$
Global ($L = 75$)	$80,9\%$	$47,4\%$	$75,0\%$
Combined ($L = 20$)	$85,0\%$	$69,2\%$	$77,2\%$
Combined ($L = 75$)	$89,4\%$	$66,7\%$	$75,0\%$

Table 1. Equivalence θ measured between the key-frames sets derived from original and processed video sequences.

4.2.2. Video feature vector representativeness

We now evaluate the aptitude of key-frames to characterize the content of a video sequence, depending on the algorithm used to detect shot boundaries, i.e. to identify video shots. First, we define the notion of *similarity* between frames. Then, we consider and compare the percentages of frames that are *similar* to the key-frames selected based on each shot boundary detection algorithm.

Given a parameter ϵ, we say that two adjacent frames are ϵ-identical if the distance measured between these two frames is smaller than $\epsilon.d_{max}$, where d_{max} is the maximum distance measured between two consecutive frames on the whole sequence. Based on this definition, the set of ϵ-similar frames associated to a key-frame r is defined as the largest set of consecutive frames that contains r and such that all pairs of adjacent frames are ϵ-identical.

Figure 3 displays the average number of frames that are ϵ-similar to a key-frame as a function of ϵ, for different shot boundary detection algorithms. Figure 3 aggregates the results for the three original video sequences introduced above. We observe that the curve corresponding to the combined approach lies above all other curves. We conclude that the shot boundary detection based on the combination of an adaptive and global threshold better capture the essence of the video sequence.

4.2.3. Video feature vector system validation

In this section, we analyze the matching between the key-frames selected in original and processed video sequences. As in Section 4.1, the matching between two frames is computed as the peak of cross-correlation (PCCs) between the RASH feature vectors extracted from the two frames. Moreover, we say that an original

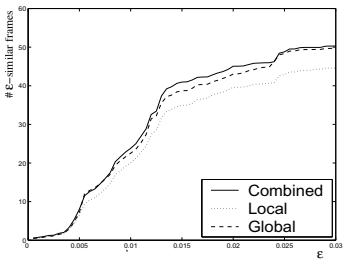


Fig. 3. Average number of video frames that are ϵ-similar to a key-frame, as a function of ϵ.

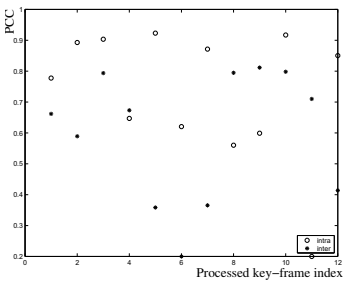


Fig. 4. Intra and Inter matching for each representative frame of the processed Monster video sequence.

sequence *corresponds* to a processed sequence if and only if the processed sequence has been derived from the original sequence. Based on these definitions, the Intra and Inter matching, associated to a key-frame selected on a processed sequence, are defined as follows. Given a key-frame selected in a processed sequence, its Intra matching is defined as its best matching with any key-frame extracted from the corresponding original sequence. On the contrary, its Inter matching is defined as the best matching with any key-frame extracted from a non-corresponding original or processed sequences (i.e. a sequence based on Swordfish or Starwars).

Figure 4 compares Intra and Inter matching for each key-frame of the processed "Monster" video sequence. About that figure, it is useful to mention that among the selected key-frames of each processed sequence, frames are labeled 8, 9, and 11 do not have an equivalent among the key-frames selected in their corresponding original sequence. Not surprisingly, the feature vectors of these "false alarm" frames present weak INTRA matching values. Unfortunately, we also observe in Figure 4 that some processed frames that do have a similar frame among the key-frames of the corresponding original result in poor Intra PCC value (e.g. the ones labelled 4 and 6). This is due to the importance of distortions between original and processed sequences (PSNRs < 25 dB!). However, and this is the most important, we observe that large cross-correlation values, i.e. PCC values > 0.8, are only achieved for Intra matching. A large PCC value can thus be used to associate a candidate sequence to the correct original sequence in the database. For completeness, note also that more complex and robust decision strategies could be imagined. An example of information that we do not exploit is the relative temporal ordering of processed and original key-frames. If an original corresponds to the processed sequence, the temporal ordering of the processed and original key-frames should be similar. The design of optimal decision strategies is left for future research.

5. CONCLUSIONS

For still images, the proposed RASH feature vector appears to be more robust, and to provide much stronger discrimination than conventional histogram feature vector. The RASH vector is thus a good candidate to build indexing systems, or content-based signature mechanisms. To take benefit from the RASH vector capabilities, video content is summarized into key-frames, each of them characterizing a video shot, and being described by its RASH vector. The resulting video hashing system works in real-time, and supports most distortions due to common image manipulations.

6. REFERENCES

[1] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication," *ICIP98*, vol. 1, pp. 435–439.

[2] C.-Y. Lin and S.-F. Chang, "Robust digital signature for image/video authentication," *ACM Multimedia*, Sept. 98.

[3] R. Venkatesan, S.M. Koon, M.H. Jakubowski, and P. Moulin, "Robust image hashing," *ICIP 2000*.

[4] J. Oostveen, T. Kalker, and J. Haitsma, "Visual hashing of digital video: applicat. and techniques," *SPIE*, 01.

[5] C.-Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. on CSVT*, February 2001.

[6] M. Johnson and K. Ramchandran, "Dither-based secure image hashing using distributed coding," *ICIP*, Sept. 03.

[7] P. K. Atrey, W.-Q. Yan, E.-C. Chang, and M. S. Kankanhalli, "A hierarchical signature scheme for robust video authentication using secret sharing," *Multimedia Modelling Conference*, 2004.

[8] F. Lefebvre, B. Macq, and J.-D. Legat, "Rash : Radon soft hash algorithm," *EURASIP*, 2002.

[9] S. Cheung and A. Zakhor, "Efficient video similarity measurement with video signature," *IEEE Trans. on CSVT*, vol. 13, no. 1, pp. 59–74, Jan. 03.

[10] A. Hanjalic, "Shot-boundary detection:unraveled and resolved?," *IEEE Trans. on CSVT*, vol. 12, no. 2, Feb. 02.

[11] R. Lienhart, "Reliable dissolve detection," *SPIE*, pp. 219–230, January 2001.

[12] H. Zhang, A. Kankanhalli, and S.W. Smoliar, "Automatic partitioning of full-motion video," *Mutimedia Systems*, vol. 1, pp. 10–23, 1993.

[13] B.-L. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. on CSVT*, vol. 5, pp. 533–544, Dec 95.

[14] P. Aigrain, H. Zhang, and D. Petkovic, "Content-based representation and retrieval of visual media: a state-of-the-art review," *Multimedia Tools and Applications*, vol. 3, Nov. 96.

[15] T. Liu, H.-J. Zhang, and F. Qi, "A novel video key-frame-extraction algorithm based on perceived motion energy model," *IEEE Trans. on CSVT*, vol. 13, no. 10, pp. 1006–1013, Oct. 03.