# ROBUST VIDEO HASH EXTRACTION

*Baris Coskun, Bulent Sankur*

Electrical and Electronic Engineering Department, Boğaziçi University, Bebek, Istanbul, Turkey
phone: +90 212 358 15 40 (ext: 1414) email: [coskubar,sankur]@boun.edu.tr
web: www.ee.boun.edu.tr

## ABSTRACT

We propose a robust video hash function for broadcast monitoring and database search applications. The method consists of binarized low-frequency components of the 3D-DCT transform of video sequences. Simulation experiments show that the perceptual hash function is unique for different video content, but that it remains invariant under selected signal processing attacks.

## 1. INTRODUCTION

The need for identification and management of video content grows proportionally to the increasing widespread availability of digital media, and in particular, digital video. The new challenge is to develop capabilities to archive, classify and retrieve video clips in a database, to identify and verify a given video, or to monitor content on broadcasts or streaming media. The one-way perceptual hashing of multimedia content is one of the well-known solutions. While the cryptographic hash function demands the exact replica of the bit string, the perceptual hash focuses similarity of content. In this sense it tolerates manipulations and modifications that leave the content similar. Most perceptual or robust hash schemes rely on the similarity of uncorrelated spectral features.

Fridrich [1] addresses the tamper control problem of still images by projecting the image blocks onto random patterns and thresholding. Venkatesan [2] extracts the image hash for indexing and database searching from the statistics of sub-band wavelet coefficients. Lefèbvre [3] uses the radon transform for a perceptual hash. Although these methods can be extended to series of images, a perceptual hash specific for a video sequence or a video clip is not much addressed in the literature.

In our work, we employ a 3D DCT-based video visual hash extraction algorithm. The algorithm consists of a normalization step followed by the hash or signature extraction step. The normalization converts given video segments into a standard spatio-temporal form. This is followed by the signature extraction where a hash sequence is extracted from the 3D DCT coefficients of the normalized sequence.

We want the video signature to possess the properties of uniqueness and robustness. A signature is robust if it does not vary when the video sequence is subjected to certain editing and signal processing operations, such as contrast enhancement or frame skipping. On the other, if the content is modified, then we expect a totally different signature. In this sense, every semantically different video sequence should possess a unique signature.

In Section 2 the normalization of video segments is presented. The hash sequence extraction is explained in Section 3. The experimental set-up and the performance are discussed in Section 4.

## 2. NORMALIZATION OF VIDEO SEGMENTS

In order to estimate a standardized video perceptual hash function it is convenient to first normalize the video sequence, that is to convert it to an equivalent sequence with a standard frame size and sequence length. This involves a series of both spatial and temporal smoothing and subsampling operations.

Let $Video(w,h;f)$, represent a video sequence by the name of Video, where w is the frame width, h is the frame height and f is the number of frames within the clip. For example, Foreman(176,144; 400) signifies the Foreman test sequence with QCIF dimensions 176x144 and with 400 frames in total. Since the essential semantic information resides in the luminance component, we perform all operations on this component. After normalizing, we denote the resulting normalized video sequence as $VideoN(.,.;.)$. In our experiments, we have chosen the target dimensions of 32x32x64 because, on the one hand, they provide an adequately concise version of the video and, on the other hand, such a reduced sequence still contains sufficient content information for a discriminating signature to be extracted. The normalization step is realized in two stages, that is, the temporal normalization and spatial normalization.

### 2.1 Temporal Normalization

The temporal smoothing extends the motion information over a larger number of frames and makes the video suitable for temporal sub-sampling. In other words, a temporal pixel tube of f-frames is considered for each of the $h \times w$ pixels in the video segment, denoted as $PT_{m,n}(i)$ $i = 1,..,f, m = 1,..,h, n = 1,..,w$. Each pixel tube is then filtered via a low-pass (averaging) box filter with kernel size $k$. The kernel size is determined based on the trade-off between robustness and uniqueness. Too large a kernel size smoothens the video excessively so as to make it look like a blurred static image. On the other hand, if too small a kernel size is chosen, motion aliasing will unfavourably impact on the extracted signature. We have found out in our experiments that $k = 20$ is a suitable value. After temporal smoothing, the video signal is temporally sub-sampled to reduce the number of frames to the target number 64.

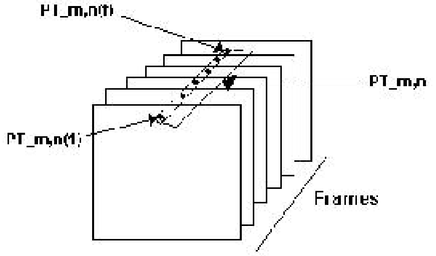

Figure. 1: Pixel Tube at the position (m,n) over f frames.

It can be argued that a motion-compensated spatio-temporal smoothing over a pixel neighbourhood can be better. In this case, the tube of each pixel neighbourhood would follow a curvilinear time trajectory over motion areas. However, motion-compensation schemes demand considerable processing power. Furthermore, we obtained satisfactory signature extraction from separate spatial and temporal smoothing.

### 2.2 Spatial Normalization

Similarly, spatial smoothing and sub-sampling are applied to reduce spatial redundancy and to extract the low-pass content. Spatial smoothing is implemented by a 2D averaging (box) filter. For the QCIF video signals used in our experiments, kernel dimensions of 7x7 proved satisfactory. Low-pass filtered frames were sub-sampled to the size 32x32.

## 3. VIDEO HASH FUNCTION

### 3.1 Hash Extraction

We consider the 3D DCT (Discrete Cosine Transform) of the normalized vide sequence VideoN(32,32,64). We expect that the 3D DCT will capture the spatio-temporal information in the frame sequence. The low-frequency DCT terms will be a robust representation of the semantic content, as they will reflect only the major changes in time or over space. Thus we select a subset of the transform coefficients $DCT\{VideoN(32,32;64)\}$. The basic trade-off of dictates on the one hand the choice of few low-pass DCT terms for robustness, and on the other hand, admission of some higher frequency terms for differentiation of close but not identical content. Our tests on video sequences have revealed that overall 64 low-pass coefficients, that is, the 4x4x4 cube in low-to-middle band are adequate for signature extraction. We excluded the lowest frequency coefficients, which can be noted as DCT(i,0,0), DCT(0,j,0) and DCT(0,0,k), since these were observed to contain little discriminatory information.

Finally these DCT coefficients were reduced to 1-bit by thresholding with respect to their median value. Any coefficient above the median value is declared as a 1 , and any below as 0 , so that we are guaranteed to have 32 1 s and 32 0 s. The one-bit quantization adds robustness to the scheme, in turn for some loss in uniqueness of the signature. On the other, equipartition of the 1 s and 0 s in the signature brings in maximum randomness on the 64-bit patterns and thus increases uniqueness.

Let the rank-ordered selected DCT coefficients be denoted as $C_{(i)}, i = 1,...,64$, for some video sequence. The median m is defined as $m = (C_{(32)} + C_{(33)})/2$. Once the median is determined, then quantization is performed as follows:

$$h_i = \begin{Bmatrix} 1 & C_{(i)} \geq m \\ 0 & C_{(i)} < m \end{Bmatrix}$$

where $h_i$ is the $i^{th}$ bit of the perceptual hash of the video signal, which is to identify a video sequence.

### 3.2 Properties of the Hash Sequences

We assume that all possible bit sequences are equally likely to occur. Recall that a hash sequence to be admissible must have equal numbers of 1s and 0s. It follows that the total number of possible hash sequences, N, is given by: $N = \binom{64}{32} = \frac{64!}{32! \times 32!} \approx 1.8326 \times 10^{18}$.

We want to calculate the Hamming distance between any two arbitrarily selected hash sequences. Among all admissible sequences, we select, without loss of generality, the special hash sequence in Fig. 2. This hash sequence has all 0s in the first 32-bit half portion and all 1s in the second half, and is used as a reference sequence for further Hamming distance calculations.

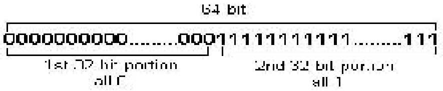

Figure 2: Special 64-bit hash pattern selected for probability calculations.

The Hamming distance between this special hash and another arbitrary selected hash is determined by the number of 1s found in the first half plus the number of 0s found in the second half. Let s in fact denote by

$(n1_1, n0_1)$ the number of 1s and 0s in the first portion, respectively, and similarly by $(n1_2, n0_2)$ those in the second half. Finally let s denote the Hamming distance with respect to the special hash sequence in Fig. 3 as $H_*$ :

$$H_* = n1_1 + n0_2 \tag{1}$$

Also, we know the following equalities: i) $n1_1 + n1_2 = 32$, ii) $n0_1 + n0_2 = 32$, iii) $n1_1 + n0_1 = 32$, and iv) $n1_2 + n0_2 = 32$. Combining equation (1) with identity (ii) and then (iii) we obtain:

$$H_n = n1_1 + 32 - n0_1 = 2n1_1 \tag{2}$$

Equation (2) states that the Hamming distance between two arbitrary hashes is always an even number, as obvious from the fact that changes in the number of 1s must be compensated from an equal number of changes in the number of 0s, and vice versa. Since the occurrence probability of a 1 bit or a 0 bit is equal, we can use the binomial model to calculate the probability distribution of $n1_1$ :

$$P(n1_1) = \binom{32}{n1_1} \times 0.5^{n1_1} \times 0.5^{(32-n1_1)} = \binom{32}{n1_1} \times 0.5^{32} \tag{3}$$

Combining equation (2) with equation (3) the Hamming distance between hashes of two arbitrary selected video clips can be written as in equation (4), which is also plotted in Figure 3.

$$P(H_*) = \binom{32}{n1_1} \times 0.5^{32} \quad , \quad \begin{matrix} 0 \le n1_1 \le 32 \\ H_* = 2 \times n1_1 \end{matrix} \tag{4}$$

Since the $P(n1_1)$ is the binomial probability function, we have the following mean and variance values: $E\{H_n\} = 2 \times E\{n1_1\} = 32$ and $\sigma^2_{H_n} = 2 \times \sigma^2_{n1_1} = 16.$

## 4. EXPERIMENTAL RESULTS

In the following experiments we prove two properties of the proposed perceptual video hash function: a) robustness, that is that the hash function does not get affected from signal processing attacks and editing effects; b) uniqueness or randomness, that is the hash sequence is clearly different for different video content. The difference of two hash sequences is measured in terms of the Hamming distance. Furthermore, in order to assess the quality of the video after attacks, the distance between a video sequence and its attacked version is measured with the Structural Similarity Index SSIM of Bovik [4,5]. The SSIM looks beyond simple pixel similarity and considers a higher-level interpretation of distortion. The SSIM goes from 0 to 1 as the similarity increases and becomes 1 for two identical images. The SSIM index of two video sequences is simply defined to be the mean of the SSIM indexes between their corresponding frames.

### 4.1 Uniqueness of the Video Hash

In our experiments, we calculated the Hamming distance between several video clips, and we observed that their distribution fits the binomial case. In Figure 3, the histogram of Hamming distances between original (unattacked) video clips is presented along with the theoretical binomial distribution. We used 45 video clips and computed 45(45-1)/2 = 990 Hamming distances. It can be observed that experimental hash values follow closely the theoretical distribution, confirming the uniqueness or randomness assumption. Notice that in this experiment the video clips were obtained from a football game sequence, that is all clips of the same genre.

### 4.2 Robustness of the Video Hash

We conducted experiments under several editing and signal processing attacks to test the robustness of the scheme. Some of the attacks were very severe in that the video sequence became almost unrecognizable, e.g., under blurring or contrast decrease, though the content was not manipulated. The resulting Hamming distances are summarized in Table 1.

The Hamming distance between the hash sequence of attacked video shot and that of its original deviates very little from zero, and in no way confounding an attacked video with another original content video. From the two distance histograms in Figure 3, a threshold value of 20 can be chosen, which fixes probability of false alarm at 0.01. The threshold ($th$) is calculated from equation (4). This threshold indicates the line of demarcation between the attacked versions of a video and a video with some other content.

The following comments can be made:

- In blurring attack, almost all the Hamming distances are below threshold value since that even very heavy blurring does not disturb low frequency DCT coefficients too much. Interestingly, contribution to Hamming distances come from plain and almost static video frames, since their low-frequency DCT coefficients are close to each other and susceptible to sign changes with small perturbations.
- AWGN attack simply superposes high frequency components on the video segment. Such perturbations are virtually unnoticed by our hash functions
- Contrast manipulation attack modifies the range of the pixel values but without changing their mutual dynamic relationship of pixels. However, extreme contrast increase results in pixel saturation to 255 and clipping to 0, which forms low frequency plain regions and consequently causes changes in the hash.
- In brightness manipulation attack, the perceptual hash is also robust. However, when brightness manipulation is taken to the extreme of saturation (too dark, clipped to 0 or too bright, saturated to 255) the hash function

suffers. Notice however this level of attack is not very realistic, as the video would loose most of its value.

- In sharpening attack, the edges are enhanced by superposing high-pass components onto video segments themselves while maintaining the low-pass components and consequently leaving perceptual hash intact.
- In frame dropping attack, the gaps between dropped frames are filled with the replicas of adjacent frames. The perceptual hash is distorted as the temporal low-pass components are distorted by replication, after which the video becomes very annoying.
- The Mpeg-4 compression basically removes the high frequency redundancy and so has very little effect on perceptual hash.

In Table 1 we give the mean ($\mu_H$) and standard deviation ($\sigma_H$) of the Hamming distances and SSIM scores. The SSIM scores indicate the strength of attacks at a level where, either the Hamming distance has reached the threshold or, the Hamming distance stays low due to robustness of the hash, but the image has suffered enough distortion to be unacceptable from SSIM point of view. The miss probability $P_M$ denotes the probability of accepting the hypothesis of a different content, while only the original content was distorted.

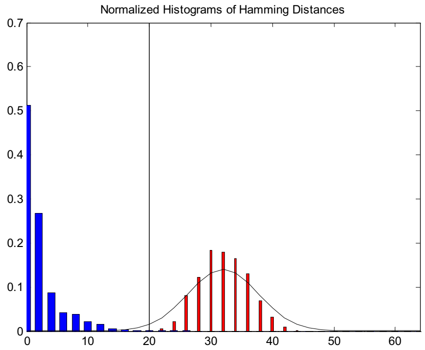

Figure 3: Histogram of Hamming distances between unattacked video clips (narrow bars); Histogram between original video and its attacked versions (wide bars). The threshold set at false alarm of 0.01.

The correlation coefficient between Hamming distance and SSIM is given in Table 2. In most of the attacks we observe very weak correlation because, as the SSIM decreases gradually, the Hamming distance still tend to stay close to zero, which is required from a perceptual hash. However, since Hamming distances rapidly increase in cases of saturation and clipping, the magnitude of correlation is high in brightness manipulation and contrast increase.

## 5. CONCLUSION

We proposed a new method for identifying video segments via short robust hashes. The hash is shown to be robust against video-processing attacks, which cause small perturbations on the video segment, but does not significantly modify the semantic content. On the other hand, the hashes of different video segments, even of the same genre and subject, yield totally different hash sequences in terms of Hamming distances. Thus the hash sequence, which is shown to be adequately unique and robust, can be used in such applications as broadcast monitoring, video database searching and etc. We pursue the evaluation of the hash function against a bigger set of attacks, and we explore alternative hash sequences, as derived from median thresholding of wavelet coefficients.

Table 1: Hamming distance for various attacks. ($P_M$ : Prob. of miss)

| **Attack** | **Hamm. Dist.** | **SSIM** | $P_M$ |
|---|---|---|---|
| | μ ( σ) | μ ( σ) | |
| *Blurring* | 1.13(1.22) | 0.53(0.16) | 0 |
| *AWGN* | 0.48 (0.97) | 0.57(0.33) | 0 |
| *Cont. Inc* | 4.11(3.90) | 0.72(0.21) | 0 |
| *Cont. Dec* | 0.26 (0.74) | 0.78(0.19) | 0 |
| *Bright. Inc.* | 2.72 (3.33) | 0.82(0.11) | 0 |
| *Bright. Dec* | 5.92 (5.62) | 0.52(0.30) | 0.014 |
| *Sharpen* | 1.88(1.29) | 0.85(0.06) | 0 |
| *Frm. Drop&Repeat* | 4.90(4.24) | 0.64(0.13) | 0.017 |
| *Mpeg4 Compress* | 0.34(0.78) | 0.76(0.02) | 0 |

Table 2: Correlation between Hamming and SSIM scores under various attacks. (CC: Correlation Coefficient)

| **Attack Type** | **CC** |
|---|---|
| *Blurring* | -0.10 |
| *AWGN* | -0.56 |
| *Contrast Increase* | -0.81 |
| *Contrast Decrease* | -0.31 |
| *Brightness Increase* | -0.73 |
| *Brightness Decrease* | -0.82 |
| *Sharpen* | -0.22 |
| *Frm. Drop&Repeat* | -0.58 |
| *Mpeg4 Compress* | -0.02 |

## REFERENCES

[1] J. Fridrich and M. Goljan., *Robust hash functions for digital watermarking. in Proceedings of the IEEE Int. Conference on Information Technology: Coding and Computing,* LasVegas, NV, USA, Mar 2000.

[2] R. Venkatesan, S. M. Koon, M.H. Jakubowski, and P. Moulin. *Robust image hashing. in Proceedings of the IEEE International Conference on Image Processing,* ICIP '00, Vancouver, Canada, Sept 2000.

[3] F.Lefèbvre, B.Macq JD.Legat, *" RASH: Radon Soft Hash algorithm ", 11th European Signal Processing Conference*, Sept 3-6 2002, Toulouse, France.

[4] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli *Image Quality Assessment: From Error Measurement to Structural Similarity IEEE Transactions On Image Processing*, vol.13, no.1, Jan 2004.

[5] Z. Wang, L. Lu, A.C. Bovik, *Video Quality Assessment Based on Structural Distortion Measurement IEEE Tran. On Image Processing*, vol.19(1), Jan 2004.