A NOVEL HASHING ALGORITHM FOR VIDEO SEQUENCES

Antonio Mucedero, Rosa Lancini and Francesco Mapelli

CEFRIEL – Politecnico di Milano
Via R.Fucini, 2 - 20133 – Milano - ITALY
E-mail: {rosa, mapelli}@cefriel.it

ABSTRACT

Recently new technologies are presented for automatic media identification based not only on watermarking. In this paper we present a new algorithm used to identify video sequences. It is based on perceptual hashing, which analyzes the signal in order to extract its fingerprint, called hash value. The idea is that this value could allow the unambiguous identification of the signal, such as human fingerprint with people. Different aspects are considered: the hash extraction process and the database strategy to retrieve stored information. For the second problem it is important to take into account the probability to not identify the video. An analytical analysis of error probability is presented. The proposed technique is tested under different kind of compression using different coding algorithm. The results show that a reliable identification can be performed even for low bit-rates.

Keywords: video fingerprinting, perceptual hashing.

1. INTRODUCTION

The increasing number of multimedia data that can be accessed increases the interest in automatic retrieval of information about that data. This can be done in different ways, but the main distinction is between techniques that actively modify the signal to add information and techniques which do not modify the data [2]. Watermarking is the main example of the first class and it is the most widespread. It hides the identification data in the original signal. Hashing – or fingerprinting – belongs to the second class [1],[3],[5]. In this case the signal is not modified but analyzed in order to extract the most important features that allow the unambiguously identification. Some approaches are proposed to use watermarking and fingerprinting together as in [4].

In this paper we propose an algorithm belonging to the class of hashing techniques. The main idea is to extract for each frame or group of frames a hash value using a mathematical analysis. The hash values are stored in a database. When it is requested to identify a video we extract its hash value and comparing it with the database we retrieve the needed information. Two main parts can be identified for the automatic video identification: the *hash value extraction* and the *database strategies*. In this paper we address mainly the first one.

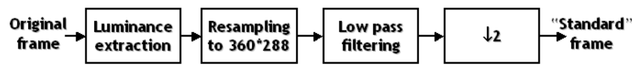

Figure 1 – Pre-processing phase.

2. HASH VALUE EXTRACTION

The hash value extraction is the first important phase in the hashing technique: we have to choose the best features that represent the video.

This task is split in two parts: *pre-processing* – which converts the original signal to a "standard" signal – and *hash computing* - to extract the hash value using robust features.

2.1. Pre-processing

In this part – shown in Figure 1 – the original signal is converted to a standard version in order to simplify the hash computing phase. In order to reduce the number of data to be analyzed, we consider the most important component for the human visual system: the luminance. This component is resampled to 352*288 pixels obtaining a fixed size to analyze. Then a low pass filter is applied and the last step is a downsampling in both directions of factor 2. Finally we have each frame represented by a 176*144 pixels luminance component.

2.2. Hash computing and storage

This part performs the extraction of the hash value from the "standard" signal. Figure 2 shows the main steps of the algorithm. They are analyzed in this paragraph.

The first phase is the v*ariance matrix construction*. A matrix containing the variance in a 15*15 block around each pixel is computed.

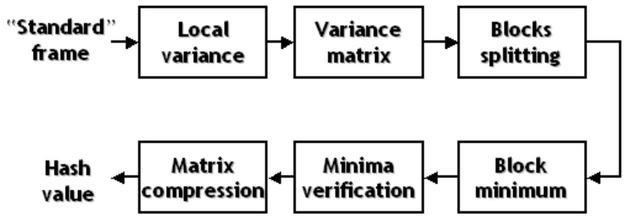

Figure 2 – Detailed hash extraction phase.

Minima number	Min block 1	Min block 2	Min block 3		Min block N

Figure 3 - Hash vector.

The second phase is the *block splitting*, which splits the variance matrix in non overlapped blocks of size 16*16. Each block is analyzed independently from the other. In this way we obtain N=(176/16)*(144/16)=99 blocks.

The next step for hash extraction is the minima selection: for each block we extract the position of the minimum – also called star. Obviously in a single block we can obtain different positions with the same value corresponding to the minimum. In this case we take the median position between the different positions of the minima.

We add an operation to verify the obtained minima: *minima verification*. If a minimum is on the block border it is not considered, i.e. we consider that the block has no minima. It is removed because under video processing they can easily move to neighbor block. The obtained matrix – called constellation –is full of "0" except in the position of a minimum where we have a "1".

Last steps of the process is the *matrix compression* – or hash storage. This operation reduces the size of the hash to be efficiently stored in a database. Figure 3 shows the stored hash: is a vector containing two kinds of data: the number of minima and their position. The position is "0" if the minimum is not present, otherwise we put its real position (1 is the upper-left corner, 256 the lower-right one). It is noticeable the very small amount of memory needed to store the vector which is 100 elements long – 99 block plus one element for the number of minima. The constellation size is not stored while it is fixed (it is a matrix of size 176*144 with blocks of 16*16 pixels). The number of minima is stored to improve the database strategy: only the constellations of frames with similar number of minima are compared.

3. DATABASE STRATEGY

Only a short description of this phase is presented because it is not the focus of our work and it can be improved using an optimized database strategy. Once extracted the hash value we need to find it in a database in order to retrieve the related information. This operation has to be performed as fast as possible. To increase speed we adopt some tricks. First of all we analyze group of frames (GOF) instead of single frames. This is obtained merging the constellation of consecutive frames: the GOF constellation is a matrix of size 176*144 obtained with an *OR* operation between the constellation of each frames of the GOF. This means that an element is "0" if and only if every constellation has "0" in the same position, otherwise we have a "1". Once analyzed a GOF we analyze the consecutive non overlapped one. Once it is identified a GOF we can refine the search to find exactly the frame if it is necessary. In our work the GOF size is 11 frames.

4. ERROR PROBABILITY ANALYSIS

To model the error probability we assume that the position of the minimum in each block can be chosen from a uniform probability. The possible positions where a minimum can be found are W^2+1, where $(W+2)^2$ is the size of a block. In fact in a block of size $(W+2)^2$, we can have up to W^2 possible positions for minima – because a minima on the border is not considered – the "1" considers the possibility of a block without minima (as explained in section 2.2). The probability to have the minimum in a particular position is:

$$p_T = \frac{1}{W^2+1}.$$

The probability to not choose the same position is:

$$p_F = \left(1 - \frac{1}{W^2+1}\right) = \frac{W^2}{W^2+1}.$$

The probability to have no errors in N blocks is:

$$P(S=0) = \left(\frac{1}{W^2+1}\right)^N.$$

The simplest way to compute the probability to find S common minima between two constellations of N blocks is:

$$P(S) = \left(\frac{1}{W^2+1}\right)^S \left(\frac{W^2}{W^2+1}\right)^{N-S} = \frac{\left(W^2\right)^{N-S}}{\left(W^2+1\right)^N};$$

which represents the false positive error probability. In practice this theoretical value is quite higher due to the fact that the complete independency is not achieved. For this reason we choose a quite low threshold value for S.

In the tests phase, the results are reported as the percentage of common minima, i.e. the value $E=S/N$. This value E represents the parameter used to check the identification. As threshold value we choose E_T=0.20: if E>0.20 we consider two frames similar, otherwise if E<0.20 we have different frames. This threshold is chosen taking into account two empirical results. First of all the number of minima in common between the same videos after compression can be quite low: we can have E=30%. Moreover the number of minima in common between different videos can be quite high: we can have E>10%.

5. TESTS

The algorithm is tested on many PAL sequences – i.e. frame size of 720x576 pixels and frame-rate of 25fps. Just as an example we compare different football matches, car races, standard sequences – like mobile and calendar – and so on. The considered attacks are the compression at different bit-rates and different standards, which are the most common operations for videos. The results are obtained extracting the hash from compressed videos (at different bit-rates) and comparing it with all hashes from all videos. The comparison is reported as the value of $E=S/N$, where S is the number of common minima between two constellations and N the total number of minima – see paragraph 4. In the following paragraph we have different values: E_m is the minimum E obtained comparing two videos, E_M is the maximum and E_A the mean. The threshold is E_T=0.20.

5.1. Comparison between original videos

The first test verifies the correct identification of frames in uncompressed video. This means that taking a random frame from a video we are able to correctly identify the video sequence and the frame position in it. In this condition each frame is correctly identified in all our tests. Obviously without any compression we have a perfect matching between video under test and original video so E=1. Comparing different videos we obtain a mean value E=0.08.

This tests highlight that the number of corresponding minima increases for frames close to the considered one (see Figure 4). This is quite obvious considering that closer we are to the reference frame more similar are frames, so more similar should be the constellation. The number of minima in common between frames of the same sequence is in general higher than the number in common for different sequences.

5.2. Tests for compressed videos

This paragraph reports some results obtained in identification of videos compressed with different techniques: MPEG-1, MPEG-2, MPEG-4 and DivX;-).

The tables 1,2,3 containing the results report in the first column the tested bit-rate, in the second one the comparison between the compressed video with its original version, while the third shows the comparison between compressed video with the other in the database. The last row of each table shows the mean of values obtained in all tests.

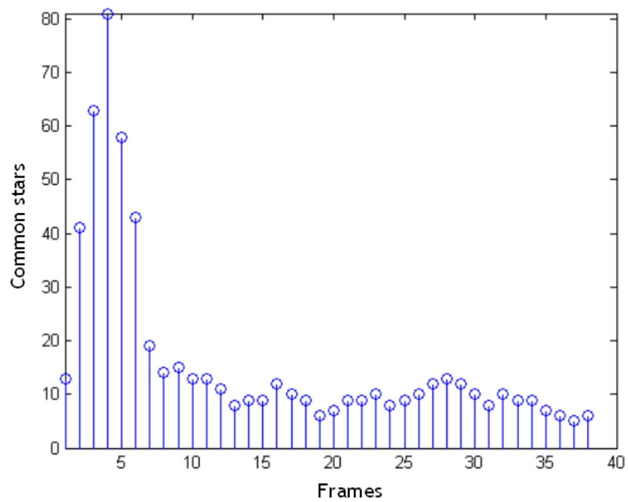

Figure 4 – Comparison between the hash of selected frames (number 4) and hash of other frames of the same video.

For *MPEG-1* compression the results are in Table 1 for different bit-rates. In this experiment the worst case is at 500kbps. We found a minimum value for similarity between two version of the same video of about E=0.28. Comparing different videos the maximum value is about E=0.12. These values compared with the chosen threshold provide good results: all videos are correctly retrieved and there are no false positive cases.

Tests for *MPEG-2* are performed in the range between 0.5Mbps to 10Mbps. The results are in Table 2. In this case the minimum value for a comparison between the original and compressed versions of the same video is E=0.23 which is above the threshold. In case of comparison between different videos the maximum value is below 0.1, which is heavily below the threshold.

Tests for *MPEG-4* and *DivX;-)* provide almost similar results. The algorithm is tested mainly for low bit-rates coding up to 1000kbps and the results are in Table 3. In this case we have the minimum value comparing the same video that is about 0.23, and the maximum for different videos comparison below 0.1. Even in this case the identification can be performed with a high reliability.

5.3. General comments on results

The results shown that the identification can be performed with a very high reliability. In our tests we do not find any false positive or false negative, which means that the threshold protects against errors and the chosen feature can be used to represent the similarity between video. The estimated error probability considering the number of tests performed is below 10^{-7}.

Bit-rate	Compressed vs. original	Different videos
0.5 Mbps	E_m=0.28 / E_A=0.52	E_M=0.12 / E_A=0.05
1 Mbps	E_m=0.30 / E_A=0.53	E_M=0.12 / E_A=0.05
1.5 Mbps	E_m=0.30 / E_A=0.55	E_M=0.12 / E_A=0.05
Mean	E_m=0.28 / E_A=0.53	E_M=0.12 / E_A=0.05

Table 1 - Results obtained for MPEG-1 compression.

Bit-rate	Compressed vs. original	Different videos
0.5 Mbps	E_m=0.23 / E_A=0.41	E_M=0.1 / E_A=0.05
1 Mbps	E_m=0.25 / E_A=0.45	E_M=0.08 / E_A=0.07
2 Mbps	E_m=0.26 / E_A=0.48	E_M=0.1 / E_A=0.04
10 Mbps	E_m=0.37 / E_A=0.75	E_M=0.1 / E_A=0.04
Mean	E_m=0.23 / E_A=0.52	E_M=0.1 / E_A=0.05

Table 2 - Results obtained for MPEG-2 compression.

Bit-rate	Compressed vs. original	Different videos
100 kbps	E_m=0.23 / E_A=0.40	E_M=0.09 / E_A=0.04
250 kbps	E_m=0.23 / E_A=0.40	E_M=0.1 / E_A=0.04
500 kbps	E_m=0.25 / E_A=0.42	E_M=0.08 / E_A=0.05
1000 kbps	E_m=0.26 / E_A=0.46	E_M=0.1 / E_A=0.05
Mean	E_m=0.23 / E_A=0.42	E_M=0.1 / E_A=0.05

Table 3 - Results obtained for DivX;-) compression.

Analyzing the numerical results two things can be highlighted:

- reducing the bit-rate lowers the similarity between the original and compressed versions of a video;
- comparison between different videos is not significantly affected from compression.

The first issue is quite obvious: bit-rate reduction causes loss of quality so the similarity with the original version lowers. This means that the chosen feature follows the human visual system behavior. The second issue is again explained with perceptual considerations: modifying the bit-rate does not modify the perceptual content of a video. This means that different videos are always different even if we reduce the bit-rate. Also in this case our algorithm has a behavior similar to the human visual system.

5.4. Comparison with other algorithm

It is difficult a numerical comparison with other techniques. For our algorithm we have a first estimation of the error probability which is below 10^{-7} – i.e. a wrong video identification every 10^{-7} comparison. This value is similar to other in literature. Regarding the implementation our approach is simpler than most of the other techniques. In conclusion we provide similar results with a quite simple approach.

6. CONCLUSIONS

In this paper we presented a novel technique for automatic identification of digital videos that can be compressed by different technologies. The algorithm is based on a hashing technique that identifies the video extracting its fingerprint. The results showed the efficiency of the algorithm at different bit-rates. We were able to identify the video even for high compression obtained using algorithm like DivX;-) and MPEG-4.

The research is now focused on different aspects to improve the algorithm. First of all we are studying a better error probability analysis. Moreover we are working on a more extensive test phase of the algorithm in order to better estimate which are the limits of the technique.

7. REFERENCES

[1] V.Fotopoulos and A.N. Skodras, "A new fingerprinting method for digital images," *Proc. 1st IEEE Balkan Conference on Signal Processing, Communications, Circuits and Systems*, Istanbul, Turkey, June 1-3, 2000.

[2] T.Kalker, J.Haitsma and J.Oostveen "Issues with Digital Watermarking and Perceptual Hashing", *SPIE Conference on Multimedia Systems & Applications*, August 2001.

[3] T.Kalker, J.Haitsma and J.Oostveen "Visual Hashing of Digital Video: applications and techniques", *SPIE Applications of Digital Image Processing,* San Diego, USA, August 2001.

[4] D.Kirovski, H.Malvar, and Y.Yacobi "Multimedia Content Screening using a Dual Watermarking and Fingerprinting System", *ACM Multimedia* 2002.

[5] R.Venkatesan, S-M.Koon, M.H.Jakubowski and P.Moulin "Robust Image Hashing", *ACM Multimedia* 2002.