ABSTRACT
Video identification is extremely important in video browsing, database search and security. In this paper, we present a video hashing based on MDS (Multi-Dimensional Scaling) which is able to work under variable video transmission impairments and resistant to signal processing. In this method, each frame of the video is divided into blocks and compute its low and middle frequency DCT coefficients of luminance component as a disparities measurement for MDS. Then the video is mapped to two-dimensional space using MDS, and generate a robust hashing as a video signature utilizing the distances between two points mapping from frames. It found that this video hashing is resistant frame geometric attacks (rotation, shift), random noises, lossy compression and other video transmission impairments. It can be instrumental in building database search, video copy detection and watermarking applications for video.

Index Terms—video signature, robust hash, video hashing, identification.

1. INTRODUCTION
Nowadays more and more digital videos are available on the web and in multimedia databases. Since there is no central management on the web, duplications of videos are inevitable. A study estimated that about 46% of all the text documents on the Web have at least one “near-duplicate” document which is identical except for low-level details such as formatting [1]. The phenomenon is likely to be more severe for videos, as they are often stored in multiple locations, compressed with different algorithms and bit rates to facilitate downloading and streaming. Similar versions, in part or as a whole, of the same video can also be found on the web when some web users modify and combine original content with their own productions. Identifying these similar contents is beneficial to many web video applications [2]. For this purpose, video hashing is used in recent years. As we all know, hash function is frequently called message digest function. Its original purpose is to extract a fixed-length bit string from a message (computer file or image) of any length. The resulting hash value is highly sensitive to every single bit of the input. These functions are extremely fragile and can not be adopted for hashing multimedia data. In multimedia hashing, it is more important to be sensitive to the content rather than the exact binary representation. For instance, the raw video, its compressed version, its low-pass filtered version, and its increased brightness version should yield the same hash value since their content is essentially the same but their binary forms are very different. So, a alternate way to compute the hash is needed for multimedia applications, where the hash function results in the same output value unless the underlying content is significantly changed [2]. In a nutshell, what’s needed in both applications discussed above is a robust hash H that sensitively on a secret key K and continuously on the video V:

1) \( H(K, V) \) is uncorrelated with \( H(K, V') \) whenever video \( V \) and \( V' \) are dissimilar;
2) \( H(K, V) \) is strongly correlated with \( H(K, V_a) \) whenever \( V \) and \( V_a \) are similar ( \( V_a \) is the video \( V \) after an attack comprising of a rotation, shift, and other signal processing);
3) \( H(K, V) \) is uncorrelated with \( H(K', V) \) for \( K \neq K' \).

There have been many robust video hashing and signature proposed in the literature. Lee[3] proposed a video signature based on the centroid of gradient orientations. It is robust against various common video processing including lossy compression and frame rate change but vulnerable to geometric transformations. Fridrich [4] addresses the tamper control problem of still images by projecting image blocks onto key based random patterns and threshold to create a robust hash, its robustness has improved but the complexity is increasing. In [5], they present a scheme that robust against frame dropping and can detect spatial cropping and temporal jittering, however, the robustness issue for compressed domain videos is not addressed. All these schemes have their own merits and weaknesses.

In this paper, we proposed a robust hashing based on MDS, which is a classic dimension reduction method. This
algorithm generates a robust hashing as a video signature utilizing the distances between two points mapping from video frames. The experimental results show that it is robust to most of signal operations and transmission impairments.

2. PROPOSED MDS-METHOD

2.1. Video Hashing Generation

The flowchart of hash generation is shown in Figure 1(a). There are three important steps in the method, which is feature extraction, MDS and hashing Calculation. We will describe them in detail.

2.1.1. Feature extraction

A robust image hash is a bit-string that somehow captures the essentials of the digital image or block, so video is. Our requirement is that we need a key-dependent function that returns the same bits or numbers from similar looking videos. As we all know, the low and middle frequency DCT coefficients are robust to signal processing operations and some geometric attacks. Meanwhile, in the model time, the multimedia information is more and more huge, when transmit these information, compression is used widely. So the video hashing should be robust through compression too. What is preserved under lossy compression? In [6], the authors explore two invariant properties of quantization based lossy compression (e.g., JPEG) for image authentication. Because each frame can be taken as an image, we extended the first invariant property to video in this paper, which is the invariant relationship between two coefficients in a block pair of two frames and it is used for the original distance measure of MDS, which is described in next part.

2.1.2. MDS

MDS is a widely used dimension-reduction method for multivariate data analysis. It finds a mapping of units in a low-dimensional space, which preserves “the intrinsic similarities” (or disparities) among all the units as faithfully as possible.

Here a classic solution of MDS is shown as follows.

- Define a distance measure between pairs of units in high dimensional space and construct a distance matrix $D$ whose elements are $d_{ij}$, then convert the matrix $D$ to $A = (a_{ij}) = (\frac{1}{2}d_{ij}^2)$.
- Construct a matrix $B = (b_{ij})$, and $b_{ij} = a_{ij} - \bar{a}_i - \bar{a}_j + \bar{a}_{**}$.

$$
\bar{a}_i = \frac{1}{n} \sum_{j=1}^{n} a_{ij}, \quad i = 1, \ldots, n
$$

$$
\bar{a}_j = \frac{1}{n} \sum_{i=1}^{n} a_{ij}, \quad j = 1, \ldots, n
$$

$$
\bar{a}_{**} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}, \quad 1 \leq i, j \leq n
$$

and $n$ is the number of units.

- Compute $k$ maximum eigenvalues of $B$ which are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ and their corresponding eigenvectors $x^{(1)}, \ldots, x^{(k)}$ that satisfied normalization conditions: $x_i^{(1)}x_i^{(k)} = \lambda_i, i = 1, 2, \cdots k$. In most time, $k$ is determined according to different requirements. We used 2 in this paper.
- Construct a $n \times k$ matrix $\hat{X} = (x^{(1)}, \ldots, x^{(k)})$ and it is a solution of MDS.

From the solution above, we also know that when distance matrix is modified, the distance between two points in the low-dimensional space may change less. This is why we use MDS in our scheme.

For a video, we can map it to a set of points in a two-dimensional space using MDS. The algorithm is given below:

**Algorithm 1: MDS For Video**

**Input**

Original video $V_o$ ($N$ frames).

**Begin**

For $k = 1: N$ Do //Loop on video frames.

Decode the video bit stream to a number of $8 \times 8$ DCT blocks, frame by frame.

Label all DCT coefficients of luminance component in zigzag order, denoted as: $\{D^o_{ij}: 1 \leq i \leq 64; 1 \leq j \leq m\}$.

$m$ is the number of blocks in one frame.
Select middle 30% ac coefficients from each block to form feature set $F_k$. Then average them and denoted this value as $k_A$.

End

For $k = 1: N$

For $m = k: N$

If $A_k > A_m$

\[ r_{km} = r_{mk} = 1 \quad // \quad R = \{ r_{ij} \}_{m \times m} \text{ is a distance matrix.} \]

Else

\[ r_{km} = r_{mk} = 0. \]

End

End

End

Take $R$ as a distance matrix, Unitize classic MDS, and map the video to a set of points. Then compute the distance between two adjacent points, denoted as $Dis = \{ Dis_i \}_{1 \leq i \leq N - 1}$.

End

Output

The distance vector $Dis$.

Figure 2 show the video and its mapping.

2.1.3 Hash Calculation

A robust hash function is the key part of the scheme. There are several hash functions in [4,7]. Here we adopt a hash function similar with that in [4]. We first generate a random vector $Q$, whose entries are uniformly distributed random variables in $[0, 1]$, then compute the mean of vector $Q$ and subtract it from each element in $Q$. Define this gotten vector as $P = \{ P_i \}$ and it is the key. Compute hash values $H = \{ H_i \}$ with a threshold $Th$ as follows:

\[
H_i = \begin{cases} 
1 & \text{if } |Dis_i \cdot P_i| \geq Th \\
0 & \text{if } |Dis_i \cdot P_i| < Th
\end{cases} \quad 1 \leq i \leq N - 1 \quad (2)
\]

For each video, a threshold $Th$ is calculated so that on average 50% of projections have absolute value larger than $Th$ and 50% are in absolute value less than $Th$. So $Th = \text{median}( |Dis_i \cdot P_i| ) \quad (3)$

2.2 Video Identification

The identification process is similar with the video hashing generation, which is shown in Figure 1(b) and given in detail bellow:

Algorithm 2: Video identification

Input

Video sender’s key $P$.

The video $V^s (N \text{ frames})$ to be authenticated, it may undergo some modifications.

Begin

For $k = 1: N$ // Loop on video frames

Decode the video bit stream to a number of $8 \times 8$ DCT blocks, frame by frame.

Label all DCT coefficients of luminance component in zigzag order, denoted as: $\{ D^w_{ij} : 1 \leq i \leq 64; 1 \leq j \leq m \}$.

$m$ is the number of blocks in one frame.

Select middle 30% ac coefficients from each block to form feature set $F^w_k$. Then average them and denoted this value as $A^w_k$.

End

For $k = 1: N$

For $m = k: N$

If $A^w_k > A^w_m$

\[ r_{km} = r_{mk} = 1 \quad // \quad R = \{ r_{ij} \}_{m \times m} \text{ is a distance matrix.} \]

Else

\[ r_{km} = r_{mk} = 0. \]

End

End

End

Take $R$ as a distance matrix, Unitize classic MDS, and map the video to a set of points. Then compute the distance between two adjacent points, denoted as $Dis = \{ Dis_i \}_{1 \leq i \leq N - 1}$.

End

Output

The Hamming Distance $ham$.

3. EXPERIMENTAL RESULTS

In this section we apply the scheme on the video “foreman.gif” (352x288, 50 frames) which is shown in Figure 2 for intra-hash statistics and “footballman.avi” (352x288, 50 frames) for inter-hash statistics. We report first the intra-hash and inter-hash distance statistics, and compared the results with those in [7].

Intra-Hash Statistics are based on the Hamming Distances of hash functions of the same video content under
various modifications. They represent the robustness performance of video hash function. If the percentage of mean hamming distance to total number is less than or equal to a threshold (many literatures use 15%), we can identify the video.

Inter-Hash Statistics are based on the Hamming Distances of hash functions of different video sequences, whether in original form or in modified forms. They represent the uniqueness, that is, discrimination performance of the video hash algorithm. It may be half of the total hash values. The detail can be seen in [7].

Figure 3 summarizes the inter-hash and intra-hash statistics. Table 1 gives comparatively the robustness performance of MDS-based scheme in this paper and scheme in [7] under the follow modifications[7].

AWGN: The high-frequency perturbation superposed on the video virtually goes unnoticed by the hash function.

Blurring: Since even heavy blurring does not much affect the low-frequency coefficients, the hash function remains very robust.

Frame rotation and shift: We consider two geometric modifications, namely frame rotation and frame circular shift.

Random frame dropping: This modification corresponds to a lossy channel when the damaged packets always coincide with the frame headers. After frame drops at randomly chosen locations, the gaps left by dropped frames are filled by linear interpolation from nearest surviving future and past frames in order to preserve the sequence length.

MPEG compression: Compression basically removes the high-frequency irrelevancy and so has very little effect on perceptual hash.

4. CONCLUSIONS

We have presented a scheme for computing a robust hash from video clips for the purposes of identification and verification. This scheme is based on MDS. We first explore an invariant feature between two frames of a video, then mapping it to points in a two dimensional space using MDS. Therefore, we get a more robust feature which is distances between two adjacent points. Finally, a set of hash values are generated utilizing a key. The proposed hash function is shown to be remarkably robust against signal-processing modifications and channel transmission impairments. In future, we will perform this scheme under more different modifications and improve its robustness.

5. ACKNOWLEDGEMENT

This work is supported by National Natural Science Foundation of China (60872024, 60970048)., National Basic Research Program of China (973 Program, No.2009CB320905), Natural Science Foundation of Shandong Province (No.Y2007G04, Q2008G03), Cultivation Fund of the Key Scientific and Technical Innovation Project (NO.708059), Ministry of Education of China , the Excellent Youth Scientist Award Foundation of Shandong Province (No. 2007BS01023). Graduate Independent Innovation Foundation of Shandong University.

6. REFERENCES