A Robust Video Hash Scheme Based on 2D-DCT Temporal Maximum Occurrence

Qian Chen, Jun Tian, and Dapeng Wu

Abstract

In this paper, we propose a video hash scheme that utilizes image hash and spatio-temporal information contained in video to generate video hash. A video clip is firstly segmented to shots, and video hash is derived in unit of shot. We notice that for video hash applications in identification and verification, reference video and suspected video always appear in pair. Therefore, we propose to derive the shot hash in a pairwise manner. For both reference and suspected shot, we apply 2-Dimensional Discrete Cosine Transform (2D-DCT) to each frame in the shot, quantize the Discrete Cosine Transform (DCT) coefficient, and record the temporal occurrence of the co-located coefficient. We then choose a pair of the closet value as the DCT coefficient for every collocated entry, inverse transform to the spatial domain, and derive image hashes from two feature frames by hash based on Radial projections (Radial hASH). Experiment results show that the proposed 2D-DCT temporal maximum occurrence (2D-DCT TMO) scheme successfully derives shot hash that represents the content, and is very robust in video identification, authentication, and verification.

Index Terms

Video hash, image hash, video authentication, video signature.

I. INTRODUCTION

Video hash techniques, also called video signature or fingerprinting, extract the most important features from a video to form compact digests that allow efficient visual content identification.
and authentication. Existing approaches of video hash can be classified into two categories. 1) Schemes in the first category extract features in spatial domain only. Usually, a robust image hash approach is first developed in this kind. Video hash is a simple extension of the image hash applied to those individual key frames extracted from the video [1][2]. 2) Schemes in the second category extract features in spatio-temporal domain. The scheme in [3] applies 3-Dimensional Discrete Cosine Transform (3D-DCT) transform to a normalized video clip, and selects a $4 \times 4 \times 4$ cube of coefficients as video hash. Since 3D-DCT contains both temporal and spatial information, this is a video hash scheme based on spatio-temporal domain. In [4], each incoming frame of a video clip is divided into small blocks. The mean luminance is calculated for each block. The hash value is the sign of the spatio-temporal difference of mean luminance. Another hash technique based on spatio-temporal color difference is reported in [5]. Each frame is divided to small blocks, and the difference between spatial and temporal similarity is used as hash function, where similarity is defined as correlation between spatial/temporal consecutive blocks.

In the spatial domain video hash technique, how to identify key frames to efficiently represent a video sequence is the most important issue. Intuitively, whatever key frame selection method is employed, key frames are only selective information of the video in both temporal and spatial domain, and they can never truly be a good global representative of the video sequence. For example, though Radial hASH works well on still images, it is not sufficiently robust in discriminating videos if simply using the image hash of key frames as video hash [1]. Meanwhile, video hash techniques in the second category extract features in spatio-temporal domain, which are completely independent solutions from key-frame based methods, as they are based on image hashing. Generally, approaches employing both spatial and temporal information tend to be more robust than key-frame based schemes because the features extracted from 3D domain are surely a better representation than 2D domain. Unfortunately, approaches reported in this category are very limited compared with the well-developed image hashing techniques [1][6][7][8][9]. Hence, we are motivated to find a better way to integrate image hash to video hash than simply applying it to some selected key frames. Besides, though 3D-DCT based video hash [3] is robust, it requires pre-processing to normalize video in spatial and temporal as to convert it to a standard cube before applying 3D-DCT. Temporal smoothing may remove minute variations of a pixel in time, but for variable length video sequences, if all temporally compressed to fixed frame
length, say \( F = 64 \), will harm the representation of the video, making it more difficult to extract meaningful features in video hash. Also, this method is very sensitive to the play order of the video, as reverse play temporally affects the DCT coefficients. Therefore, we are interested to improve 3D-DCT based video hash by solving the problems.

We propose a robust video hash scheme based on 2D-DCT temporal maximum occurrence (2D-DCT TMO). It extracts spatio-temporal information. We observe that to extract features directly from a complete video sequence is very difficult. While a video shot, i.e., a group of frames of continuous action in time and space, is a more meaningful unit for video processing and identification. Therefore, we first divide a video sequence to multiple shots. We then apply 2D-DCT to each frame in a shot, quantize the DCT coefficient, and record the temporal occurrence of the co-located coefficient. We select those DCT coefficients of the maximum occurrence and do inverse DCT transform back to spatial domain to construct a representative frame, and calculate the image hash for this frame, set the frame hash as video hash for this shot. Hence, video hash is actually the shot hash in our scheme.

The rest of the paper is organized as follows. In section II, some basic procedures before deriving video hash in 2D-DCT TMO is introduced, including shot segmentation and DCT coefficient selection. Section DCT-TMO2 gives details of 2D-DCT TMO, explaining how to obtain shot hash in pairwise manner to 2D-DCT maximum occurrence. Section IV presents the experimental results and discusses its performance. And section V concludes the paper.

II. SOME BASIC PROCEDURES BEFORE 2D-DCT TMO

In our method, a video sequence is first divided into several video shots. Each shot consists of a group of frames of continuous action in time and space. And then we spatially normalize shot to a pre-defined WxH size. Unlike in [3], we do not temporally normalize a video sequence to V(W,H,F) 3D cube, where \((W,H)\) and \(F\) are respective size on spatial and temporal dimensions after subsampling. On the other hand, we also do not do temporal normalization to the segmented video shot. Because we propose to count the number of temporal occurrence of spatial features (2D-DCT histogram), and each frame will be a valuable sample in statistics.
A. Video Shot Segmentation

The goal of video shot segmentation (video shot detection) is to find significant disparities between consecutive frames of the sequence. In our scheme, we want to localize the shot boundary as accurate as possible so that all frames in a shot display similar temporal and spatial features. Also we want the shot segmentation to be stable, i.e., the detected shot boundary will not be affected by video length. Any video shot detection method that meets the requirements above can be used for shot segmentation. In this paper, we employ a shot detection based on regional histogram difference [10]. A frame is divided to certain number of blocks, say $4 \times 4 = 16$ blocks, compute the gray level histogram difference between collocated blocks of two consecutive frames, and count those blocks with the difference metric greater than a certain threshold $T H_1$. If the number of such blocks exceeds a pre-defined value $T H_2$, we decide this frame is a shot boundary frame. From our experiment, we find $T H_1 = 0.1$ and $T H_2 = 12$ shows quite good performance.

B. DCT Coefficient Selection

The DCT transforms an image from spatial to frequency domain. As most of the energy is now concentrated in low-frequency region, it forms a much more compact representation of the image than in spatial pixel domain. That is, the amount of information contained in a few DCT coefficients may be close to all $M \times N$ pixel values. Hence, we choose 2D-DCT coefficients as the features extracted from spatial domain. As mentioned before, we normalize each frame to a pre-defined spatial scale, which we choose to be QCIF (176) size in this paper. Then we need to figure out how many 2D-DCT coefficients are enough for a QCIF size image. Fig. 1 shows a typical DCT energy spectrum, i.e., DCT variance distribution along with the DCT index in zig-zag scan order. It is based on a randomly picked video clip from Big Bunny Buck of 5000 frame length. Obviously, variance, as an energy metric, decreases sharply as DCT index grows. While small index number indicates low frequency component, and high index for high frequency, which can be guaranteed in zig-zag scan order. Fig. 2 shows in a more straightforward way of energy distribution over DCT index(frequency component). The 3 curves in the figure represents 3 randomly picked video clips extracted from Big Bunny Buck of 5000 frame length each. They are of the similar shape. It is seen that most of the energy compacts in the first few DCT coefficients (low frequency components) in DCT domain. Fig. 3 gives a closer look at the
energy ratio of the first 100 DCT coefficients. As a tradeoff between accuracy and complexity, we select the first 28 DCT coefficients in zig-zag order, which takes up around 70% of all energy, as the extracted spatial feature. This number is also used in [11] without explanation.

III. SHOT HASH BASED ON 2D-DCT TEMPORAL MAXIMUM OCCURRENCE

The first 28 DCT coefficients in DCT domain of each frame are good spatial features. As all video frames in a video shot have similar visual content, the collocated DCT coefficients along the temporal order will also show similar values, and a value or values (if several values
occur in equal frequency) that are most likely to occur can reflect the features of the entire video shot. To derive the temporal feature of a video shot, we count the number of occurrence of the collocated DCT coefficients in a video shot in a histogram manner. For example, assume a video shot has $T$ frames, and all first 28 DCT coefficients are carefully quantized. Among $T$ DCT coefficients marked Number 1 in zig-zag order, we do histogram statistic, and re-order the histogram in descent order to find the DCT value/values most likely to occur. Likewise, we copy this procedure to all remaining 27 DCT coefficients in these $T$ frames. This procedure has been demonstrated in Fig. 4.
Fig. 5. Quantization Table of the selected 28 DCT coefficients.

Intuitively, for each of the 28 DCT coefficient entry, a single value that occurs most frequently should be able to represent the entire video shot. However, we cannot simply pick one most frequently occurring value, due to the following reasons:

1) If there are several values of equal occurrence, to pick any one of them will inevitably lose the accuracy of temporal information. No single value has priority over others if they all appear the same times given the frame number.

2) Because of the post-processing to video shot, such as geometric manipulation, temporal manipulation and compression etc., the one DCT value having most occurrence in such video shot may differs from the true value in the original video shot. How much it differs from the true value depends on the process itself, but from our experiment, such difference does exist in most of the video shots inspected.

3) The importance of DCT coefficients differs. Low frequency components contain more energy and information [11], and should be assigned a larger number of values to choose from.

To this end, we design a quantization table for 28 DCT coefficient entries based on the one in [11] as in Fig. 5. The number in each entry is the quantization level for that DCT coefficient, for instance, the first DCT is equally quantized to 120 levels, the second and third DCT are quantized to 70 levels, etc. Note that the smaller DCT index, the greater quantization level reserved for that entry, which indicates the importance of each DCT coefficient.

As mentioned in Section I, the application of video hash is efficient visual content identifi-
cation, video database search etc. These applications have one thing in common - it involves at least two video contents, one suspect video and one reference video. All the existing video hash methods derive video hash independently for both videos and measure the distance between the two hashes to decide the content similarity of the two videos. It guarantees there is one unique hash for each video. Instead, in this paper, we propose a novel scheme that derives video hash in a pairwise manner, i.e., no hash for an individual video can be obtained without referring to another video. Obviously, this breaks the unique property of the video hash, since different video hashes will be derived from the same video content referring to different references. However, the proposed pairwise manner shows excellent performance in all applications in Section IV. Fig. 6 illustrates the flow chart of the pairwise shot hashes derivation. After quantizing each DCT entry based on the quantization table and reorder them in descending order for the individual shot, we need to collaboratively select a pair of coefficients for each collocated DCT entry from the reserved candidate levels in a shot pair. A simple way is to always choose a pair of collocated DCT coefficients that are closest to each other. Regarding that closer DCT values indicate more similarity in spatial domain, we guarantee that the pairwise shot hashes always indicate the most content similarity. Hence, if the distance metric between such two shot hashes is still large, they are very likely to have completely different contents.

Furthermore, we point out that to choose the closest value DCT pair may not always be a good choice. For example, if there is an exact same DCT value in collocated DCT entry in both shots, but in reference shot it occurs very frequently, while in suspect shot it is very rare, such DCT value should be excluded from the candidate level since we prefer DCT levels with high temporal occurrence in both shots. Therefore, among all the quantization levels in descending occurrence order for each DCT entry (a frequency spectrum), only those taking up the first half of the energy in the frequency spectrum will be reserved as candidates. We do such screening process for both shots so that only good temporal representative DCT levels are kept and selected in pairwise manner for shot hashes. Fig. 7 shows the frequency spectra of the first 3 DCT entries after reordered quantization for reference shot and suspect shot respectively (the 5th shot from Big Bunny Buck). When restricting the selection range within the first half energy of each spectrum, only a few DCT levels of the highest temporal occurrence will be kept, as shown in Table I. We then choose a pair of DCT levels of closest values for each collocated DCT entry. Finally, we have a pair of 28 DCT coefficients for both reference and suspect shots.
In our proposed 2D-DCT TMO scheme, the 28 DCT coefficients are not the final shot hash. We have to inverse transform the DCT coefficients and derive image hash from the spatial domain as the ultimate shot hash. It is termed in this paper as feature frame for a shot. This is based on the following considerations:

1) The structure of the 28 DCT coefficients between any two shots is quite similar, with one large value of the first DCT coefficient (DCT 1) and the rest are much smaller values compared to DCT 1. This structure makes it difficult to distinguish two shot hashes because the one large value of DCT 1 makes the others all look like 0.
TABLE I

NUMBER OF DCT LEVELS KEPT FOR THE FIRST 3 DCT ENTRIES IN REFERENCE AND SUSPECTED SHOTS

<table>
<thead>
<tr>
<th></th>
<th>DCT 1</th>
<th>DCT 2</th>
<th>DCT 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference shot</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Suspected shot</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

2) As mentioned before, we are motivated to find a better way to integrate the well developed image hash methods to video hash. It is expected that the more similarity the two shots have in content, the more visually alike of the spatial pictures. Given the fact that the distance between image hashes depends on the difference of the image contents, we can use the image hash of the feature frame as the shot hash. Any image hash method that is unique and robust can be implemented. In this paper, we employ the Radial hASH in [1], and compute the cross correlation of feature frame hashes as the distance measure.

IV. EXPERIMENTAL RESULTS

In this section, we report the identification and verification performance of the proposed 2D-DCT TMO video hash scheme.

The identification problem is defined as the ability to recognize a video shot (reference shot) from video clip database (collection of suspect shots). What makes it difficult is the suspect shot may have been attacked or modified and differs from the true version in reference.

In our experiment, we take a 10 min length video clip from movie *Big Buck Bunny* as the test video. Certain shots are extracted as reference. Suspect video, which is the full 10 min length video, will firstly be segmented to shots, and we try to identify the reference shots from the the entire suspect video clip. For a given reference shot, we compute the shot hash in a pairwise manner with each shot in the suspect video clip, and identify the one with the highest cross correlation value as the match shot. Table II presents partial results of the identification experiment. We notice that cross correlation peaks at exactly 1 if the reference and suspect shots do have the same content. Fig. 8 gives the feature frames of the reference shot and its matching suspect shot of Shot 14 and Shot 22. It can be seen that they are exactly the same, and therefore...
Fig. 7. Frequency spectra of the first 3 DCT entries for reference shot (original video) and suspected shot (compressed video).
(a) Reference shot: DCT 1 (b) Reference shot: DCT 2 (c) Reference shot: DCT 3 (d) Suspected shot: DCT 1 (e) Suspected shot: DCT 2 (f) Suspected shot: DCT 3

the image hashes derived from such frames should be identical. Meanwhile, we find the cross correlations between two shot pairs, Shot 14 & Shot 16 and Shot 42 & Shot 44 are very close to 1. Likewise, Fig. 9 shows the feature frames of the two pairs, which are highly alike visually. And we also observe that the contents of the two shots, including the figure and the background of the shot pair does appear very high similarity, which are shown in Fig. 10. Fig. 10(a) and Fig. 10(b) are frames extracted from shot 14 and 16 respectively, while Fig. 10(c) and Fig. 10(d) are from shot 42 and 44. Consequently, our proposed 2D-DCT TMO can successfully identify shot from the database, and the derived shot hash does represent the content of the shot. The more content similarity of the shot pair, the higher cross correlation between shot hashes.

The verification problem, on the other hand, is defined as the effort to prove or disprove if a video clip is what it is claimed to be. The suspect video usually is altered, post-processed or
Fig. 8. Feature frames for the reference shot and its matching suspected shot of Shot 14 and Shot 22. (a) Feature frame for reference shot 14 (b) Feature frame for suspected shot 14 (c) Feature frame for reference shot 22 (d) Feature frame for suspected shot 22

Fig. 9. Feature frames for two non-matching shot pairs with high cross correlation. (a) Feature frame for reference shot 14 (b) Feature frame for suspected shot 16 (c) Feature frame for reference shot 42 (d) Feature frame for suspected shot 44
TABLE II
PARTIAL CROSS CORRELATION RESULTS FROM IDENTIFICATION EXPERIMENT (R: REFERENCE SHOT NUMBER, S: SUSPECTED SHOT NUMBER)

<table>
<thead>
<tr>
<th></th>
<th>S14</th>
<th>S16</th>
<th>S22</th>
<th>S42</th>
<th>S44</th>
</tr>
</thead>
<tbody>
<tr>
<td>R14</td>
<td>1.00</td>
<td>0.9994</td>
<td>-0.6126</td>
<td>0.4759</td>
<td>0.4876</td>
</tr>
<tr>
<td>R16</td>
<td>0.9994</td>
<td>1.00</td>
<td>-0.6241</td>
<td>0.4526</td>
<td>0.4685</td>
</tr>
<tr>
<td>R22</td>
<td>-0.6126</td>
<td>-0.6241</td>
<td>1.00</td>
<td>-0.1215</td>
<td>-0.1571</td>
</tr>
<tr>
<td>R42</td>
<td>0.4759</td>
<td>0.4526</td>
<td>-0.1215</td>
<td>1.00</td>
<td>0.9990</td>
</tr>
<tr>
<td>R44</td>
<td>0.4876</td>
<td>0.4685</td>
<td>-0.1571</td>
<td>0.9990</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Fig. 10. Extracted frames from two shot pairs that appear similar content in Big Buck Bunny. (a) Frames from Shot 14 (b) Frames from Shot 16 (c) Frames from Shot 42 (d) Frames from Shot 44

modified in various ways before the verification test. Hence, a robust video hash is necessary in order to tolerate the malicious attacks or accidental manipulations to the video. These changes include but not subjected to geometric transform, blurring, frame drop, time clipping, compression and noise interference. The performance of verification depends on the cross correlation between the shot hashes of the suspect shot and the reference shot. A predefined threshold of cross correlation is set to decide either to accept or reject the claimed content. In our experiment,
TABLE III

VERIFICATION PERFORMANCE

<table>
<thead>
<tr>
<th>Methods of modification</th>
<th>Cross correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift(15 pixels)</td>
<td>0.9759</td>
</tr>
<tr>
<td>Rotate(20 pixels)</td>
<td>0.9989</td>
</tr>
<tr>
<td>Resize(10 pixels)</td>
<td>0.9939</td>
</tr>
<tr>
<td>Frame drop</td>
<td>0.9999</td>
</tr>
<tr>
<td>Time clipping</td>
<td>0.9994</td>
</tr>
<tr>
<td>Contrast change</td>
<td>0.9938</td>
</tr>
<tr>
<td>Blurring</td>
<td>1.00</td>
</tr>
<tr>
<td>AWGN</td>
<td>0.9999</td>
</tr>
<tr>
<td>Compression</td>
<td>0.9999</td>
</tr>
</tbody>
</table>

we purposely manipulate an original shot content (Shot 22 in *Big Buck Bunny*) in different ways to create various versions of suspect shots, and compare these shot hashes with the reference shot hash of the original content. These manipulations are geometric transform (shift, rotate and resize), frame drop (periodically shrink to 1.5 times shorter of the original length), time clipping (cut shot to 40% original length), contrast change, blurring, through AWGN channel, and video compression (H.264/AVC). The result is shown in Table III. Note that the cross correlation between the shot hashes of the altered version and the original shot is no lower than 0.97, which verifies the robustness of the proposed hash scheme. In fact, extensive tests over sequences all show the similar result. Therefore, we conclude that 0.97 is a good threshold to decide whether two shots are visually similar or not.

V. CONCLUSION

This paper proposes a robust video hash scheme based on 2D-DCT temporal maximum occurrence. The video clip is firstly segmented to shots, and video hash is actually derived in unit of shot. We notice that for video hash applications in identification and verification, reference video and suspect video appear in pair. Therefore, we propose to derive the shot hash in a pairwise manner. For both reference and suspect shot, we count the number of quantized DCT coefficients in each of the 28 selected DCT entry, only keep those of the highest temporal
occurrence, and choose a pair of the closest value as the DCT coefficient for every collocated entry. We then do inverse transform to the reserved DCT coefficients for both shots, and derive image hashes from two feature frames by Radial hASH. Experiment results show that the proposed 2D-DCT TMO successfully derives shot hash that represents the content, and is very robust in video identification, authentication, and verification applications.

REFERENCES