SHARING VIDEO ANNOTATIONS

Yaron Caspi
Institute of Computer Science
Hebrew University of Jerusalem
Jerusalem 91904 Isreal
+972-8-9380913
caspi@012.net.il

David Bargeron
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
425-703-7526
davemb@microsoft.com

ABSTRACT

This paper describes an approach for locating annotations generated in one video and properly placing them in a second, modified version of the same video. We focus on modifications that standard Television (TV) broadcasts may experience, including content insertion and deletion (i.e., commercials), format conversions, and different start/end recording times. To overcome these modifications, we propose content-based video timelines. We identify the position of a frame in a long video stream with an accuracy of one frame based on its content, without using embedded time codes. To make this approach feasible, we use a compact representation of a video frame which we call a "fingerprint." Fingerprints capture small temporal variations within shots, and therefore allow precise position recovery. Our fingerprints' efficient storage size, extraction time, and comparison complexity suggest that our approach can be applied using off-the-shelf PCs and TV "set-top" boxes.

1. INTRODUCTION

Video annotation has been proposed and implemented in restricted domains such as E-learning, and mechanisms for generating and sharing such annotations have been proposed [1]. Few of these systems scale well given the characteristics of commercial Television (TV). This is because the "same" TV content may undergo subtle modifications throughout its lifetime, which can render frame numbers or embedded time codes unreliable as means for identifying positions along the content's timeline. Such modifications may include insertion and deletion of short clips (i.e., different sets of commercials); format conversions (e.g., NTSC to PAL); changes in compression format and/or compression parameters; and time compression (i.e., selective dropping of frames).

Applications such as providing DVD-like chapter indexes for regular TV programming demand a more robust means of identifying positions in TV content, and this report addresses that challenge. Given two versions of the "same" TV program (i.e., that a human viewer would identify as being essentially the same, even though they may differ somewhat in content, format, etc.), the goal is to properly place annotations generated for one version of the program in the second version, using only the pixel intensity values available in the video frames themselves. For this task we propose *content-based timelines* composed of a sequence of "fingerprints." Each fingerprint is an iconic 0/1 bitmap which captures the spatial division of a single frame image into dark and bright regions. This representation is simple to construct, efficient to browse, and it provides robust and reliable positioning accuracy.

1.1 Related Work

We are not aware of any previous attempts to export and import annotations originated from TV. However, the basic challenge addressed in this paper — finding a short clip from one version of a video in a modified version of the same video — bears similarities to other applications, notably video indexing, watermarking, and video copy protection.

The typical video indexing task attempts to summarize a video using a few key frames (e.g., [3,7,8,9]). Each key frame represents a visually coherent video segment (a "shot"). Thus the extracted features have to discriminate between different shots, yet be resistant to temporal variation within a shot. A common approach is to exploit statistics of light (e.g., color histograms). Such statistics ignore the spatial information in a video frame, and as we shall see, they are inappropriate for support of the fine-grained discrimination required for accurate positioning among the frames *within* a shot.

Another area where a high level of discrimination among similar images is required is in digital rights management. In contrast to widely-used "watermarking" — wherein identification information is embedded in the media — we

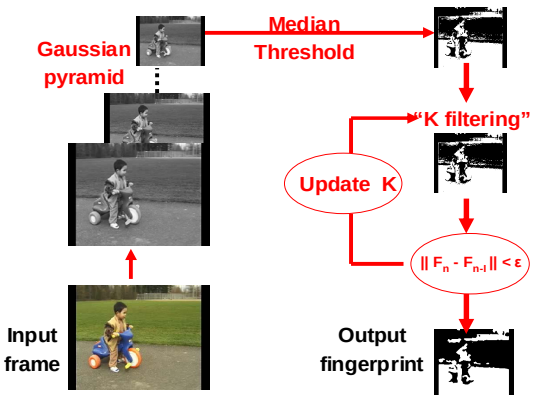

Figure 1: Fingerprint extraction algorithm

are interested in "content-based watermarking." Mihçak and Venkatesan describe such an approach for single images [5]. They extract a low-dimensional "signature" from an image they wish to protect, and then they embed identifying information into the extracted signature. We use a similar compact representation; however we do not embed the extra identifying information.

Finally, a variety of work has focused on video copy detection [4]. Here, the challenge is to recognize whether a given video program is a copy of another. With the exception of Oostveen et al [6], who base their method on temporal frame differences, most approaches do not address the temporal aspect of video, and instead rely on a comparison of key-frames. Our method differs in technical approach, class of allowed modifications (we allow temporal modifications such as format conversions or time compression, while others implicitly assume synchronization), and the nature of accompanying applications we support.

2. VIDEO FINGERPRINTS

The basic building blocks of our representation are frame *fingerprints*. These are low-resolution 0/1 binary images describing dark and bright regions in the original frame. A block diagram of our fingerprint extraction algorithm is shown in Figure 1. Given a color video frame, we consider only intensity, and we rescale it to standard SIF dimensions (320x240 pixels). We then construct a Gaussian pyramid and produce a low-resolution 40x30 pixel image. This image is thresholded using the median gray level in the frame. The resulting 0/1 bitmap has roughly the same number of 1's and 0's. The equilibrium of 1's and 0's provides maximal entropy with respect to other possible thresholds. This contributes to the discrimination of the resulting fingerprints.

Iterations of "K-filtering" (a type of morphological filtering) are then applied to remove noise. Each pixel is set to 1 if more than k pixels are 1 in a 3x3 window around it (otherwise set to 0), and we update the value of k on each iteration such that the equilibrium of 1's and 0's is maintained: If the number of 1's decreases, the value of k is decreased; if the number of 1's grows, the value of k is increased. K-filtering stops when the number of pixels updated is small (in our implementation this threshold is 5). Typically only a few iterations are needed.

We define the distance between any two fingerprints (F_1, F_2) as their Hamming distance (the number of bits that differ).

We also approximate the a priori reliability of a fingerprint using the percentage of pixels that are unlikely to change their value in a modified version of that frame. This is the percentage of pixels with gray levels that are *not* close to the median gray level:

$$(1)\ Reliability(F) = \left(1 - \frac{\sum_{m-n}^{m+n} hist(F)}{\#\ pixels}\right) \times 100\%$$

where m is the median value, and n is the noise level (in our experiments we use n=2). Based on empirical observations we found that when reliability drops below 90%, the probability of identifying the same fingerprint after modifications reduces substantially. Thus we only use frames with a priori reliability of over 90%.

3. PROPERTIES OF VIDEO FINGERPRINTS

Discrimination

Figure 2 on the next page illustrates the typical distances between corresponding and non-corresponding fingerprints of frames taken from two versions of a video, where the versions differed by compression ratio, format, and some content. The distance between corresponding frames was 2.5%-5% (15-30 bits), while the average *minimal* distance of a frame that is not present in the video to all frames in the video was 45%. Thus our fingerprints discriminate fairly well.

Size and Complexity

The size of a single full-sized fingerprint is 150 bytes, compared to 512 bytes for a typical histogram. Fingerprint construction complexity is dominated by the complexity of constructing the Gaussian pyramid (85%) — which is similar in complexity to constructing a color histogram — however fingerprints are much more efficient for browsing than histograms, since they require only simple bit comparisons. On a Pentium III workstation running Microsoft Windows XP, fingerprints can be computed in real time (e.g. at video-rate), and the fingerprints corresponding to an hour's worth of video (30x60x60=108,000 frames) can be linearly searched in under 1 second.

Robustness

Fingerprints are robust to global color modifications such as contrast and gamma correction because of the way median thresholding is employed in their construction. To

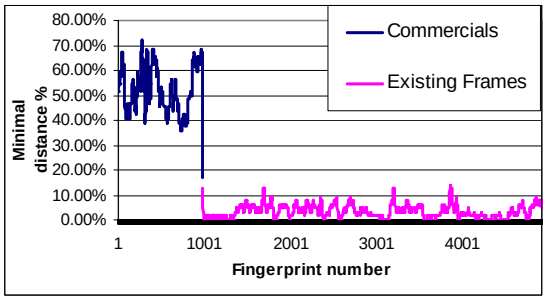

Figure 2: Feature bitmaps typical distance

Distances between the fingerprints of one movie and the best-match fingerprints from a modified version of the movie. Blue data points represent frames that exist in the modified movie but not in the original version (which simulates inserted commercials).

test their robustness against local spectral modifications, we compared the distance between the fingerprints of frames from a high quality (uncompressed) movie, and matching frames from versions of the same movie that were compressed using different levels of Indeo 5.1 compression. The average distance over 7000 fingerprints was under 6 bits (0.5%) for all compression levels we tested.

Fingerprints are also robust against some spatial transformations. For instance, small unknown spatial transformations are corrected as a result of the subsampling in the construction of the Gaussian pyramid during fingerprint creation, and robustness against large known spatial transformations results from rescaling each frame to fixed (SIF) dimensions.

Spatial Features vs statistical features

We compared the discriminative characteristics of fingerprints to that of histograms for the particular requirements of robust, precise positioning. Figure 3 displays the distance of the fingerprint of the i^{th} frame from the fingerprint of the first frame in a home video of a mother rocking a baby (essentially a single, long shot), compared to the distance of corresponding histograms under various similarity measures. The shallow slopes of the histogram graphs indicate that color histograms are highly similar, regardless of the similarity measure used. In contrast, the distance between fingerprints grows rapidly, indicating more temporal discriminative strength. This indicates that the spatial features of fingerprints are more appropriate when high temporal precision is required.

4. SEARCHING FINGERPRINTS

In order to locate the frame in a video to which an annotation belongs, we must search through the corresponding fingerprints for the one that most closely matches the fingerprint stored with the annotation.

A major challenge when searching for a single frame in a video is recognizing false detections. That is, we want to avoid reporting a positive match if the shot to which the

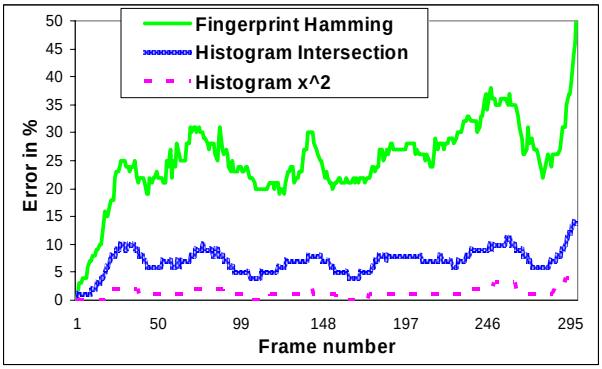

Figure 3: Spatial vs statistical features

*Comparison of the evolution of distances between fingerprints with distances between histograms in a video. All distances are from the first frame in the video. The rapid growth of distances between fingerprints (top curve) with respect to the moderate growth for histogram shows that fingerprints are more suitable for discrimination **within** a shot.*

frame belongs is missing from the video we are searching. The shot may be missing for a variety of reasons, for instance because the user started viewing/recording the video too late, or stopped too early; or there may be deletions, for instance of advertising content; or frames may have been dropped as the result of a conversion process. Figure 2 illustrates empirically that the distance between fingerprints of frames from two arbitrarily-chosen shots is almost always above 40%. During searches, therefore, we use the fingerprint of the single frame we are searching for to determine whether the shot to which it belongs is *present* in the video.

Locating the frame in the video with an accuracy of one frame, however, requires a bit more work. This is because frames that appear in the same shot are often too similar to be individually distinguishable, and this is reflected in their fingerprints (see Figure 3). Thus, to accurately locate a single frame, we match a sequence of fingerprints (for instance, 25 consecutive fingerprints) centered on the frame we are interested in. We call this sequence a "query clip."

Finding query clips

First, we sum the distances of all fingerprints in the query clip judged against possible locations in the video (and weighted by each fingerprint's a priori reliability). Only locations where the frame might be present are considered, as determined by a fast initial search across the whole video. Locations in the video with small weighted sum (e.g., less than 15%) are denoted as candidate locations and further analyzed. There may be several candidate locations.

To further refine the match, we use dynamic programming. The goal is to find a minimal global score that combines the distances between fingerprints with a penalty for missing frames. The penalty term reflects prior knowledge about the modifications that commercial TV

programs may undergo, including format conversions (e.g., PAL/NTSC, or 2:3 pull-down used in Telecine) and time compression (a method used in TV broadcasting to conserve time). Both are achieved by dropping frames in a fixed pattern, therefore our penalty term prefers jumps with fixed, regular gaps. The candidate location with lowest global score is selected as the matching segment.

4.1 Efficient search.

In a one-hour movie, there are about ~100,000 frames, 99% of which are typically pruned using a single fingerprint. Comparing a single pair of fingerprints may require as many as 2x1200 operations, so search time is dominated by the computation of Hamming distances. To reduce this complexity, we apply two shortcuts:

Data reduction

We only compare the 128 highest-variance bits in each fingerprint. We compute a mask describing the variance of each bit across a set of fingerprints by summing bit values along the time axis. High bit variance corresponds to a sum which is close to half of the total number of frames. The 128 highest-variance bits in the mask are the ones we choose from each fingerprint to compare. This approach was motivated by work done in the audio domain using Principal Component Analysis (PCA) [2]. The problem with regular PCA is that it does not preserve the 0/1 bit-representation. Our method of choosing pixels with high variance does not take into account the high order statistics that PCA does, but it does preserve the 0/1 representation, it can be implemented very efficiently, and it is effective in practice.

Skipping frames

The second shortcut we employ is based on the observation that the distance between consecutive fingerprints in the timeline is small (See Figure 3). Using the triangle inequality, we know that if the distance between a query clip ***A*** and a location ***B*** in a movie is large, and the distance between location ***B*** and another location ***C*** is small, then the distance between ***A*** and ***C*** must be large. Thus we do not need to compare the query clip to location ***C***.

To exploit this idea, as part of our content-based timeline representation for a whole video we store a pointer from each frame to the next frame that has a Hamming distance of more than 10% from it. During searches, we can skip all frames in-between.

These shortcuts help reduce the running time of searches by a factor of almost 40 (10 from data reduction, and 4 from skipping frames).

5. EVALUATION

To illustrate the wide applicability of our approach, we tested it against four types of TV programs: a sports broadcast, a sitcom, a dramatic movie, and a concatenation of short clips. In each test, our algorithm attempted to locate query clips in a modified version of a program, where the query clips were taken from an unmodified version of the same program. Each query clip was 1 sec (25 frames), and there were a few hundred query clips in each test. The modifications between the two versions of each program included different compression algorithms, bits rates, time compression (regularly dropped frames), deleted shots, errors in the Telecine process, and analogue broadcasting noise ("scintillations"). Results are summarized in Table 1. Note that in all tests, query clips were located with 1-frame accuracy at least 85% of the time, and with 1-sec (25 frame) accuracy at least 98% of the time. No false negatives or false positives occurred.

Movie Type	Perfect Match	Accuracy 1 sec	Error 1 sec or more	False Detections
Sports	96.0%	4.0%	0.0%	0
Sitcom	96.5%	3.4%	0.1%	0
Movie	97.8%	2.0%	0.2%	0
Clips	85.0%	13.0%	2.0%	0

Table 1 Summary of detection results

We implemented an email-based annotation sharing application to further explore the effectiveness of our approach. Using the application, a user can send an email message that contains an annotation plus 25 fingerprints identifying the location in a video where the annotation belongs. A recipient who opens the message sees the annotation automatically popup in the right position in the video in less than one second.

6. REFERENCES

[1] Bargeron, D., Grudin, J., Gupta, A., Sanocki, E., Li, F., and LeeTiernan, S., "*Asynchronous Collaboration Around Multimedia Applied to On-Demand Education.*" Journal of Management Information Systems. Vol 18, No 4, Spring 2002, p. 117-146.

[2] C. Burges, J. Platt, and S. Jana, "Extracting noise-robust features from audio data," *Proc. of ICASSP*, Orlando, FL, 2002, pp. 1021-1024.

[3] U. Gargi, S. Antani, and R. Kasturi, "Performance Characterization and Comparison of Video Indexing Algorithms," *In CVPR*, Santa Barbara, CA, 1998, pp. 559-565.

[4] A. Hampapur and R. Bolle, "Comparison of Distance Measures for Video Copy Detection", IBM Research Report, May 2001.

[5] M. Mihçak and R. Venkatesan, "New Iterative Geometric Methods for Robust Perceptual Image Hashing," *Proc. of ACM Workshop on Security and Privacy in Digital Rights Management*, Philadelphia, PA, Nov. 2001, pp. 13-21.

[6] J. Oostveen, A. Kalker, and J. Haitsma, "Visual Hashing of Digital Video: Applications and Techniques", *Proc. of SPIE applications of digital image processing 24*, July 2001, San Diego, USA.

[7] J. Smith, *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis*. Ph.D. thesis, Columbia University, Feb. 1997.

[8] VideoQ: www.ctr.columbia.edu:/VideoQ

[9] www.altavista.com/video.