

Automatic Identification of Digital Video based on Shot-level Sequence Matching

Jian Zhou

Department of Electrical and Computer Engineering
Ryerson University
350 Victoria Street, Toronto, Ontario
Canada, M5B 2K3
jzhou@ee.ryerson.ca

Xiao-Ping Zhang

Department of Electrical and Computer Engineering
Ryerson University
350 Victoria Street, Toronto, Ontario
Canada, M5B 2K3
xzhang@ee.ryerson.ca

ABSTRACT

To locate a video clip in large collections is very important for retrieval applications, especially for digital rights management. In this paper, we present a novel technique for automatic identification of digital video. This new algorithm is based on dynamic programming that fully uses the temporal dimension to measure the similarity between two video sequences. A normalized chromaticity histogram is used as a feature which is illumination-invariant. Dynamic programming is applied on shot-level to find the optimal nonlinear mapping between video sequences. Two new normalized distance measures are presented for video sequence matching. One measure is based on the normalization of the optimal path found by dynamic programming. The other measure combines both the visual features and the temporal information. Experimental results show that the shot-level approach is robust to frame rate conversion, color correction, and compressions. The proposed distance measures are suitable for variable-length comparisons.

Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems- *Video*; H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing.

General Terms

Algorithms, Design, Experimentation, Performance, Theory.

Keywords

Video similarity measure, video identification, dynamic programming, shot-level video sequence matching.

1. INTRODUCTION

Content analysis of video is to extract meaningful information such that efficient classification, indexing, retrieval, and filtering are possible. One crucial step for such tasks is to define a similarity/dissimilarity measure between two video sequences.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'05, November 6–11, 2005, Singapore.

Copyright 2005 ACM 1-59593-044-2/05/0011...\$5.00.

The common techniques rely on key-frames since classical methods developed in content based image retrieval can be applied on these still-frames. In [1], a fast video signature based on randomized algorithms is proposed to approximate the video similarity defined as the percentage of clusters of similar frames shared between two video sequences. In [2], block-based minimum variances are used to create video hash values. However, temporal information is ignored in both of the above methods. A template-frequency model which makes uses of the temporal dimension is proposed in [3]. Another similarity measure between shots is developed by using dominant color histograms and spatial structure histograms [4]. In [5], the similarity between the query image and the video is defined as the distance between the query point and the linearly interpolated feature line. However, it is observed that video similarity measure is essentially a multiple-to-multiple matching process. For example, the query is not necessarily one key-frame or one shot. A query containing multiple frames or even multiple shots is also possible. One of the few research works that consider such a scenario is presented in [1]. However, the sequences are not treated as ordered sets since the frames are randomly sampled from video sequences. Therefore, the algorithm does not distinguish two sequences such as “AABBCC” and “CCBBAA”. Also, if the video database contains many similar video sequences, the method proposed in [1] might not have enough discrimination abilities. Examples include sports video such as soccer video and football video. Note that key-frame based methods are not suitable for such query tasks since most of the scenes in those videos are very similar. Some other applications, such as digital rights management, also require quick identification of nearly the same content. Therefore, it is often necessary to incorporate order and temporal information. The desirable distance should be a proximity measure between two ordered sets.

In this paper, we present a shot-level video similarity measure based on dynamic programming. Note the temporal information such as shot durations is not affected by frame rate conversion or illumination changes. The proposed method can be used to locate and identify a video sequence in large collections. Unlike the technique in [6] [7], where a frame-level dynamic programming is used to deal with frame misalignment, our new method uses shot-level dynamic programming, where shot sequences are created in an illumination-invariant color space by clustering video frames in independent component analysis (ICA) subspace. In addition, two new normalized distances are introduced to calculate the dissimilarity. Optimal path is found by dynamic programming.

The presented new method is robust to histogram processing, and frame rate conversion. The new distance measures are insensitive to the lengths of videos.

2. VIDEO SIMILARITY MODEL

There is a growing concern about the digital video piracy. Due to its digital nature, the content with almost the same quality can be easily copied, edited, transformed, and distributed. Finding a specific video among large collections is very important for digital rights management applications as well as the similarity-based video retrievals. For example, a movie clip may be edited and converted to another file. Many specific attributes, such as frame rate, compression format, aspect ratio, color correction scheme, might have changed. During video editing, some inappropriate shots could be deleted and commercial breaks could be inserted. However, from human perception, we still regard them as the same content. Thus, in order to identify a specific video, an efficient video similarity method is required to identify the same content. Most existing similarity models are not suitable for such tasks since they either ignore the temporal dimension, or simplify the query model. In the presented method, new video similarity models based on dynamic programming are developed. We integrate both visual features and shot durations into dynamic programming framework, allowing variable-length comparison and partial matching.

2.1 Shot Detection

The first step is to segment a video into a shot sequences using a method in our previous work [8], where illumination-invariant chromaticity histograms are used as raw features and an ICA based method is used to convert the 256-dimensional raw feature subspace into a two dimensional feature space, in which a dynamic clustering algorithm is employed to cluster video frames into shots.

2.2 Shot-level Feature Extraction

The normalized chromaticity histogram is selected as a shot-level visual feature. The illumination-invariant normalized chromaticity (r, g) [9] is defined as: $r=R/(R+G+B)$, $g=G/(R+G+B)$. Histograms with 256 bins are generated in the normalized chromaticity color space for each frame of the video. During implementation, only r component is used for simplicity. Each shot is represented by a feature vector which is the mean vector of all video frames within the same shot. A shot sequence is then a vector sequence, $\{\mathbf{r}(i), i=1, \dots, N_R\}$, where $\mathbf{r}(i)$ represents the i -th shot and N_R is the total number of shots. Shot lengths (measured in time) are also calculated during feature extraction.

2.3 Similarity Model

Let $R = \{\mathbf{r}(1) \ \mathbf{r}(2) \ \dots \ \mathbf{r}(N_R)\}$ be a reference shot sequence of length N_R and $T = \{\mathbf{t}(1) \ \mathbf{t}(2) \ \dots \ \mathbf{t}(N_T)\}$ be a test shot sequence of length N_T . In general, the number of shots in R is not equal to the number of shots in T , i.e. $N_R \neq N_T$. Denote the two alignment functions (shot index functions) by $p(\cdot)$ ($1 \leq p(i) \leq N_R$) and $q(\cdot)$ ($1 \leq q(i) \leq N_T$) for R and T respectively. The pair of alignment functions forms an ordered set which is defined as a path [10]. The overall cost D is defined as:

$$D = \sum_{i=1}^{N_p} d(\mathbf{r}(p(i)), \mathbf{t}(q(i))), \quad (1)$$

where N_p is the total length of the path and $d(\cdot)$ is the distance measure which needs to be carefully designed to measure the dissimilarity between two feature vectors. The optimization goal is to find the alignment functions $p(\cdot)$ and $q(\cdot)$ that minimize the overall cost D in (1).

To design a suitable distance measure $d(\cdot)$, the feature vectors need to be properly scaled or normalized such that all features contribute equally. Cosine measure is the cosine of the angle between two vectors. This measure captures a scale-invariant similarity. The distance function $d(\cdot)$ based on cosine measure is defined as:

$$d(\mathbf{r}(i), \mathbf{t}(j)) = 1 - \frac{\mathbf{r}(i)^T \cdot \mathbf{t}(j)}{\|\mathbf{r}(i)\|_2 \cdot \|\mathbf{t}(j)\|_2}. \quad (2)$$

The dynamic programming can be employed for shot sequence comparison since the cost is additive, i.e., according to Bellman's optimality principle, we have the following recursive equation:

$$D_{\min}(p(i), q(i)) = \min_{p(i-1), q(i-1)} [D_{\min}(p(i-1), q(i-1)) + d(p(i), q(i) | p(i-1), q(i-1))]. \quad (3)$$

Constraints including global constraints, local constraints, and end point constraints are given as:

- (a) $p(1) = 1, q(1) = 1$;
- (b) $p(N_p) = N_R, q(N_p) = N_T$;
- (c) $0 \leq p(i) - p(i-1) \leq 1; 0 \leq q(i) - q(i-1) \leq 1, \forall i \geq 1$;
- (d) $p(i) - p(i-1) + q(i) - q(i-1) \geq 1$.

The constraints defined above guarantee the alignment paths are monotonically non-decreasing.

2.4 Normalized Distance Measure

The overall cost D can be used to measure the distance or dissimilarity between two video sequences. A desirable property for such a measurement is that the cost D should not depend on the lengths of the sequences. Therefore, a proper normalization of the total cost is necessary. For string matching, the problem has been addressed in [11] using *normalized edit distance*. However, it is computationally expensive. In practice, D/N_p can be used to calculate the distance measure with a certain amount of normalization. Our first new simplified normalization measure D_1 is defined as:

$$D_1 = D_0 / N_p, \quad (4)$$

where D_0 denotes the original total cost, i.e., $D_0 = D_{\min}$, with D_{\min} defined in (3).

For video sequence comparison, normalization of the total cost by the length of the path is essentially related to the number of shots since the length of the path is bounded between $\max(N_R, N_T)$ and $(N_R + N_T)$. Note that one video sequence with more shots does not necessarily imply it is longer than the other. However, in terms of

video similarity measure, people are often interested in how long the two video sequences “overlap” instead of how many shots (or key-frames) are similar. In another word, if we have two pairs of dissimilar shots, it is reasonable to penalize the longer sequences more, compared with the other pair with relatively shorter durations. Therefore, the *normalized edit distance* proposed in [11] cannot be directly applied here since it is only penalizes the lengths of sequences without considering the duration of each symbol. We present the second new distance measure to integrate both visual features and shot durations for video sequence comparison as follows:

$$D_2 = \frac{\sum_{i=1}^{N_p} [d(\mathbf{r}(p(i)), \mathbf{t}(q(i))) \cdot |L_R(p(i)) - L_T(q(i))|]}{\sum_{i=1}^{N_p} [L_R(p(i)) + L_T(q(i))]}, \quad (5)$$

where $L_R(n)$ is the duration for n -th shot in R and $L_T(n)$ is the duration for n -th shot in T . It is easy to show that D_2 has an upper bound as follows:

$$D_2 \leq \frac{\sum_{i=1}^{N_p} \max(L_R(p(i)), L_T(q(i)))}{\sum_{i=1}^{N_p} (L_R(p(i)) + L_T(q(i)))}. \quad (6)$$

This new distance measure D_2 combines both visual feature and time information. For applications that do not require strong temporal information, the distance measure D_1 can be used. While all the distances defined above can be used to measure the distance between two video sequences, the original total cost D_0 highly depends on the length of the path. For D_1 , the values are within the range of $[0, 1]$ since the cost is normalized by the length of the path. However, to compare large video sequences, even if the two sequences are very dissimilar, the value of D_1 may still be very small because of the large length of the path. That makes it difficult to evaluate the variable-length comparisons or choose a suitable global threshold to identify videos. On the other hand, the value of distance measure D_2 is numerically stable and at the same time has good discrimination ability, as will be shown by the numerical results.

3. EXPERIMENTAL RESULTS

To show the effectiveness of the proposed algorithm, a one hour movie (drama) is arbitrarily captured as a test example. The movie video is encoded in MPEG-2 in NTSC format with a frame rate of 29.97, frame size of 352x240, aspect ratio 4:3, and bit rate 3249kbps VBR. The video is then manually divided into several smaller clips (A'-G') as reference videos. Based on these reference video clips, MPEG-1 test video clips (A-G) are created with a frame rate of 24, bit rate 1411 kbps, aspect ratio 16:9, and a simple color correction. Therefore, totally 14 video clips are used in our tests. By applying shot detection algorithm on each clip, fourteen shot sequences are created. Dynamic programming is used to find the optimal alignment path. Three distance measures D_0 , D_1 , and D_2 are calculated and the results are shown in Table 1, 2, and 3 respectively. Though the values on the diagonal are relatively small for all three distances, Table 1 shows that D_0 does relate to the number of shots. For example, all values in the fourth

row and the fourth column in Table 1 are relatively large. That is because both shot sequences D and D' have more shots than the others. In practice, for example, if we get a distance measure 1.30, we cannot decide if the same content has been identified since that number might mean “similar” between long sequences but “dissimilar” between short sequences. The improved distance D_1 normalizes the cost by path. As we mentioned earlier, this measure is not consistent with human perception because temporal information is not considered. Hence, D_1 can be used when temporal information is not important for some applications. Table 3 shows the results for the proposed distance D_2 . As can be seen, the values are numerically stable (see (6)) and provide a consistent normalization. In practical applications, a global threshold could be easily selected to identify the video.

Table 1: Original total cost (D_0).

	A'	B'	C'	D'	E'	F'	G'
A	0.87	2.54	2.28	5.60	1.69	2.85	2.22
B	3.10	1.18	2.21	5.87	2.31	4.22	3.02
C	2.31	2.63	0.42	4.72	1.91	3.62	1.56
D	6.93	6.34	5.18	1.30	4.87	10.2	3.63
E	2.13	3.17	2.33	5.13	0.17	3.44	2.20
F	3.00	3.69	3.96	10.6	3.33	0.13	3.15
G	2.62	3.13	2.05	4.04	1.90	3.08	0.49

Table 2: Total cost normalized by path (D_1).

	A'	B'	C'	D'	E'	F'	G'
A	0.22	0.36	0.46	0.40	0.42	0.71	0.44
B	0.62	0.17	0.37	0.59	0.33	0.60	0.43
C	0.46	0.44	0.07	0.34	0.32	0.60	0.22
D	0.53	0.63	0.37	0.09	0.44	0.73	0.26
E	0.53	0.45	0.39	0.47	0.04	0.86	0.44
F	0.75	0.53	0.66	0.76	0.83	0.04	0.63
G	0.65	0.45	0.34	0.29	0.38	0.62	0.10

Table 3: Proposed distance measure (D_2).

	A'	B'	C'	D'	E'	F'	G'
A	0.01	0.10	0.37	0.35	0.23	0.08	0.41
B	0.43	0.04	0.30	0.15	0.20	0.22	0.26
C	0.36	0.23	0.02	0.25	0.22	0.09	0.54
D	0.33	0.09	0.25	0.04	0.25	0.11	0.50
E	0.20	0.15	0.21	0.22	0.01	0.07	0.32
F	0.07	0.27	0.07	0.10	0.06	0.01	0.09
G	0.31	0.26	0.33	0.54	0.34	0.09	0.01

In the second experiment, we further specifically show that the proposed dissimilarity measure especially the distance D_2 is effective to correctly evaluate the variable-length comparisons when other distances fail. We select a video segment A' from the reference video as the query, and four video segments A_s , A_l , B_s , and B_l from the test video as our test dataset. Note that a test video and a reference video differ in frame-rate, aspect ratio, and lighting conditions. The video clips are not of the same length and each contains different numbers of shots. Semantically, the test clip A_s is a subset of A' , and A' is a subset of A_l , while B_s and B_l have no overlapping with A' . We use the null set symbol ϕ to denote this no overlapping relationship. A_s and B_s are smaller video clips, compared with A_l and B_l . The relationship between the test dataset and the query video A' is listed in the second column in Table 4. The dissimilarity measures between A' and each of the clips in the test dataset are computed and listed in Table 5. Intuitively, A' should have relatively small distances with A_s and A_l , but large distances with B_s and B_l . However, as it can be seen in Table 5, the results show that D_0 and D_1 cannot reflect the true semantic relationship, since the measures are affected by the length and the number of shots. But D_2 is still able to identify that A_s and A_l are more similar to A' , compared with others.

Table 4. Data used for the second experiment.

	# of shots	relationship with A'
A'	19	
A_s	14	$A_s \subset A'$
A_l	43	$A' \subset A_l$
B_s	10	$A' \cap B_s = \phi$
B_l	30	$A' \cap B_l = \phi$

Table 5. Results for the second experiment

	D_0	D_1	D_2
A' vs. A_s	2.112	0.111	0.067
A' vs. A_l	11.094	0.258	0.179
A' vs. B_s	5.413	0.257	0.398
A' vs. B_l	12.747	0.425	0.342

4. CONCLUSION

In this paper, we present a novel technique to identify video clips based on content. The algorithm operating on shot-level fully makes use of the temporal information. A video similarity model

which combines both visual features and shot durations is presented. The nonlinear optimal mapping between the reference video and the test video is achieved by using dynamic programming. Experimental results show that the method is robust to frame rate conversion, histogram level editing, and compression format. In the presented framework, we develop new distance measures for video sequence comparison. The proposed video distances are numerically stable and consistent with human perception. Other potential applications and our future work include content based video retrieval among collections with high similarities and commercial breaks detection.

5. REFERENCES

- [1] S.-S. Cheung and A. Zakhor. Efficient video similarity measurement with video signature. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 59-74, Jan. 2003.
- [2] R. Lancini, F. Mapelli, and A. Mucedero. Automatic identification of compressed video. In *Proc. ICASSP'04*, vol. 3, pp. 445-448, May 2004.
- [3] P. Muneesawang and L. Guan. Automatic relevance feedback for video retrieval. In *Proc. ICASSP'03*, vol. 3, pp. 1-4, Apr. 2003.
- [4] T. Lin, C.W. Ngo, H.J. Zhang, and Q.Y. Shi. Integrating color and spatial features for content-based video retrieval. In *Proc. ICIP'01*, Oct. 2001.
- [5] L. Zhao, W. Qi, S.Z. Li, S.Q. Yang, and H.J. Zhang. Content-based retrieval of video shot using the improved nearest feature line method. In *Proc. ICASSP'01*, vol. 3, pp. 1625-1628, May 2001.
- [6] H. Cheng. Temporal registration of video sequences. In *Proc. ICASSP'03*, Apr. 2003.
- [7] Y. Tan, S. Kulkarni, and P. Ramadge. A Framework for Measuring Video Similarity and Its Application to Video Query by Example. In *Proc. ICIP'99*, pp. 106-110, Kobe, Japan, Oct. 1999.
- [8] J. Zhou and X.-P. Zhang. Video shot boundary detection using independent component analysis. In *Proc. ICASSP'05*, Mar. 2005.
- [9] M.S. Drew, J. Wei, and Z.-N. Li. Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalize images. *ICCV'98*, pp. 533-540, 1998.
- [10] S. Theodoridis and K. Koutroumbas. *Pattern Recognition (2nd edition)*. Academic Press, 2003.
- [11] A. Marzal and E. Vidal. Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 926-932, Sept. 1993.