A Novel Scheme for Video Similarity Detection

Chu-Hong Hoi, Wei Wang, and Michael R. Lyu

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
{chhoi,wwang,lyu}@cse.cuhk.edu.hk

Abstract. In this paper, a new two-phase scheme for video similarity detection is proposed. For each video sequence, we extract two kinds of signatures with different granularities: coarse and fine. Coarse signature is based on the Pyramid Density Histogram (PDH) technique and fine signature is based on the Nearest Feature Trajectory (NFT) technique. In the first phase, most of unrelated video data are filtered out with respect to the similarity measure of the coarse signature. In the second phase, the query video example is compared with the results of the first phase according to the similarity measure of the fine signature. Different from the conventional nearest neighbor comparison, our NFT based similarity measurement method well incorporates the temporal order of video sequences. Experimental results show that our scheme achieves better quality results than the conventional approach.

1 Introduction

With the rapid development of compute networks and Internet, digital videos become more and more easily copied and distributed. How to fast and effectively search similar video copies among huge volume database has attracted more and more focuses recently [1, 2, 3, 4, 5]. Two major applications of video similarity detection are video copyright issue and video retrieval by a given sample [6].

More and more copyright problems have been aroused as digital video data can easily be copied, modified and broadcasted over the Internet. Although digital watermarking provides a possible solution, it may not be suitable in every case. Video similarity detection has been proposed as a good complementary approach of digital watermarking for the copyright issues [3].

Furthermore, content-based video retrieval has been considered an important and challenging task in multimedia domain. Seeking an effective similarity measurement metric is regarded as a significant step in content-based video retrieval [7].

In this paper, we propose a novel scheme for video similarity detection. The rest of this paper is organized as follows. Section 2 discusses challenges of video similarity detection and related work. Section 3 presents our framework for video similarity detection and related contents are briefly discussed. Section 4 proposes

E. M. Bakker et al. (Eds.): CIVR 2003, LNCS 2728, pp. 373–382, 2003.

the coarse similarity measure scheme. Section 5 presents the fine similarity measure scheme. Finally, Section 6 provides our experimental results and Section 7 gives the conclusions and future work.

2 Challenges and Related Work

It is a challenging task to fast and effectively search similar videos from large video databases. Several papers have addressed how to tackle the problem [2, 3, 4, 9]. In general, two major research efforts for content-based video similarity detection are feature representation techniques and similarity measurement methods. Effective feature representation for video content is the first key step toward similarity detection. The second step is to find effective similarity measurement methods for cost-efficient similarity detection. We focus on the research work of the second step in this paper.

Although many efforts have addressed the problem, it is still difficult to solve the problem effectively and efficiently. Naphade et al. provide a video sequence matching scheme based on compacted histogram in [2]. A. Hampapur et al. examine several distance measurement methods and compare their performances in [3]. R. Mohan presents a scheme for video sequence matching based on similarity of temporal activity in [9]. However, most of them are based on the key-frame comparison of video shots to measure the similarity. None of them carefully consider the temporal order of video sequences and the efficiency problem. In [4], S.C. Cheung et al. develop an efficient randomized algorithm to search similar video. Their idea is based on generating a set of frames which are most similar to a set of randomly seeded frames. Although the algorithm is efficient, the idea of random order and frame-based comparison do not exploit the temporal order among video sequences as well.

3 A Two-Phase Similarity Detection Framework

Toward the challenging issue of fast and effective similar video detection from vast video databases, we propose a two-phase similarity detection framework based on different granular signatures, shown in Fig.1. In the preprocessing step, the low level features of the query video example and compared video data are first extracted. Based on the low level features, we generate two kinds of signatures with different granularities for each video sequence. Coarse signatures are generated based on the density histogram of feature points by mapping the original data space to a new pyramid space [10], while fine signatures are obtained by generating simplified feature trajectories of video sequences. In the first phase, most of statistically unrelated video data are fast filtered out by coarse similarity measure based on the Pyramid Density Histogram technique. In the second phase, fine similarity measure is performed by computing the similarity of feature trajectories of the video sequences based on the result set of the first phase. Different from the conventional approach, our fine similarity measurement method based on feature trajectories thoroughly considers the temporal order of video sequences. In the following sections, we discuss these techniques in detail.

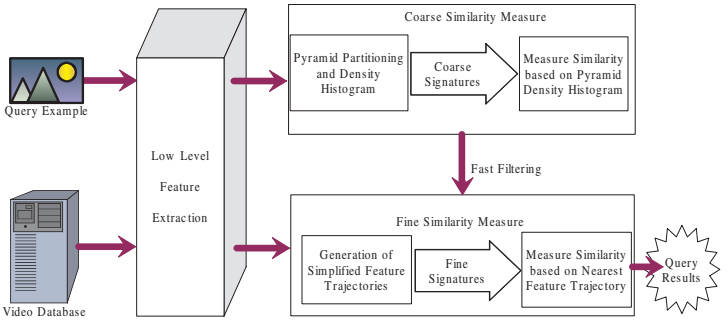

Fig. 1. A two-phase similarity detection framework

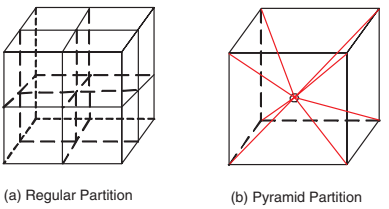

Fig. 2. Partitioning of the high dimension data space

4 Coarse Similarity Measure

Based on our proposed framework, each frame of a video sequence is considered as a feature point in the original data space after the feature extraction. Thus, a video sequence is formed by a set of feature points in the high dimension data space. For efficient similarity measure, it is impossible to conduct the similarity measurement frame-by-frame of video sequences. Therefore, we propose a Pyramid Density Histogram (PDH) technique to fast filter out the unrelated video sequences.

4.1 Pyramid Partitioning and Density Histogram

Pyramid partitioning technique is first proposed for dimension reduction and indexing problems in [10]. For a d-dimension data space, instead of infeasible regular partitioning of Fig. 2(a), the pyramid partitioning technique splits the data space into $2d$ pyramids with a center point $(0.5, 0.5, ..., 0.5)$ as their top and a $(d-1)$-dimension hyperplane of the data space as their base, shown in Fig. 2(b).

Suppose a video sequence S is formed by M frames corresponding to M feature points with $d-$dimension. Each feature point v in the video sequence S is denoted as $v = (v_1, v_2, ..., v_d)$, where $0 \leq v_i \leq 1$. Based on the pyramid partitioning technique, for a given feature point v, we assign v to the i-th pyramid by following the condition below

$$i = \begin{cases} j_{max}, & \text{if } (v_{j_{max}} < 0.5) \\ j_{max} + d, & \text{if } (v_{j_{max}} \geq 0.5) \end{cases} \quad (1)$$

where $j_{max} = \{j | (\forall k, 0 \leq (j,k) < d, j \neq k : |0.5 - v_j| \geq |0.5 - v_k|)\}$. The height of point v in the i-th pyramid is defined as [10]

$$h_v = |0.5 - v_{iMODd}|. \quad (2)$$

For each feature point v in the video sequence S, we can locate it in a unique pyramid. By computing the distribution of feature points in each pyramid, we propose the PDH technique to map the video sequence S in the original data space to the new pyramid data space. Two kinds of PDH techniques are engaged: Naïve Pyramid Density Histogram and Fuzzy Pyramid Density Histogram.

4.2 Naïve Pyramid Density Histogram

By applying the basic pyramid partitioning technique to density histogram, we present the original pyramid density histogram called Naïve Pyramid Density Histogram (NPDH). Given a video sequence S, the NPDH vector of S is denoted as $u = (u_1, u_2, ..., u_{2d})$. For each point v in S, the NPDH vector u is iteratively updated as

$$u_i = u_i + h_v \quad (3)$$

where i is defined in Eq.(1) and h_v is defined in Eq.(2). After processing all points in video sequence S, we obtain the NPDH vector as a coarse signature for video sequence S.

4.3 Fuzzy Pyramid Density Histogram

From NPDH, we found that it cannot fully exploit all information in each dimension. Thus, we propose another alternative technique called Fuzzy Pyramid Density Histogram (FPDH). For each point v in a video sequence S, instead being completely allocated to a unique pyramid in NPDH, the point v is assigned to d different pyramids based on the value of each dimension of v. The FPDH vector u is thus calculated as below

$$u_i = u_i + h_v \quad (4)$$

$$i = \begin{cases} j, & \text{if } (v_j < 0.5) \\ j + d, & \text{if } (v_j \geq 0.5) \end{cases} \quad (5)$$

where j=1,2,...,d and h_v is defined in Eq.(2). Performance comparison result of FPDH and NPDH is shown in Section 6.

4.4 Coarse Similarity Measure Based on PDH

Based on the PDH technique, each video sequence is mapped to a $2d$-dimension feature vector as a coarse signature in the pyramid data space. We then conduct the coarse filtering based on the coarse signatures. Suppose u_q is the PDH vector for a query example Q and u_c is the PDH vector for a compared video sample C in a database. Let us denote by ε a threshold for the coarse similarity filtering. Then we conduct the coarse filtering based on the comparison result of the Euclidean distance of two vector u_q and u_c. That means the compared video C is filtered out if the following condition is satisfied

$$||u_q - u_c|| > \varepsilon. \tag{6}$$

Based on the PDH technique, we can perform the coarse filtering very fast and obtain a small subset of original video database. In order to improve the precision rate, we need to make a further fine similarity measurement in the second phase.

5 Fine Similarity Measure

Although there remains a small subset of compared samples in the second phase, it is still infeasible to perform the similarity measure with the frame-by-frame comparison. Considering the temporal order of video sequences, we propose a Nearest Feature Trajectory (NFT) technique for effective similarity measurement. Instead of regarding a video sequence as a set of isolated key-frames in the conventional ways, we consider the video sequence as a series of feature trajectories formed by continuous feature lines. Each feature trajectory reflects a meaningful shot or several shots with gradual transition. Different from the conventional key-frame based comparison, our proposed similarity measure based on the nearest feature trajectories of video sequences can well exploit the temporal order of video sequences and obtain more precise results.

Nearest Feature Line (NFL) technique is first proposed for audio retrieval in [11]. It is also proved to be effective in shot retrieval of video sequence in [12]. In here, we use the similar technique to solve the similarity detection issue. Different from the NFL used in [12], our proposed NFT scheme consider the global similarity measurement of feature trajectories in two video sequences. A feature trajectory in our scheme is formed by a lot of continuous feature lines. Different from the Simple Breakpoint (SBP) algorithm used in [12], we propose a more effective algorithm to generate the simplified feature trajectories.

5.1 Generation of Simplified Feature Trajectories

As we know, each frame in a video sequence is considered as a feature point in the feature space. Two neighboring feature points form a feature line. A lot of feature lines in a shot form a feature trajectory. Thus a video sequence can be represented by a series of feature trajectories called a fine signature. However, it is impractical to regard all frames in the video sequence as the feature trajectory

for the efficiency problem. Thus, we propose an efficient algorithm to generate the simplified trajectory.

Given a video sequence, we first detect the hard cut transitions of shots. For each shot, we generate a simplified feature trajectory as follow. Suppose we have a shot S and the number of frames in the shot is N, denoted as $S = \{v(t_1), v(t_2), ..., v(t_N)\}$. And let us denote by S' the simplified feature trajectory and denote by N^{ψ} the number of frames in S', which is a subset of S. However, it is time-consuming to obtain an global optimum subset S'. Therefore, we propose the following effective algorithm which can achieve a local optimum answer.

Assume that $\{v_k | k = 1, 2, ..., N\}$ represent the frames in a video sequence. Let us denote by $LR(v_k)$ the local similarity measure function of point v_k. Then, we define the following similarity measure function

$$LR(v_k) = |d(v_k, v_{k-1}) + d(v_{k+1}, v_k) - d(v_{k+1}, v_{k-1})| \tag{7}$$

where $d(v_i, v_j)$ means the distance between point v_i and point v_j. Obviously, v_{k-1}, v_k and v_{k+1} satisfy the triangle-inequality relation. In the special case, if $LR(v_k)$ is equal to 0, then point v_k is on the line of points v_{k-1} and v_{k+1}. That means the variance of trajectory at point v_k can be neglected; otherwise v_k deviates from the line of points v_{k-1} and v_{k+1}. Apparently, the larger the value of $LR(v_k)$ is, the larger the deviation of the trajectory at that point is. After computing the $LR(v_k)$ value of each point, we remove the point whose value of $LR(v_k)$ is the minimum of all points. Repeat the procedure until the number of remaining points in the simplified trajectory is equal to N^{ψ}.

5.2 Similarity Measure Based on the Nearest Feature Trajectory

Based on the fine signatures discussed above, we proposed an effective algorithm for fine similarity measure of video sequences. Given two video sequences, the dissimilarity measure focuses on measuring the similarity distance of different feature trajectories. In the following part, we focus how to formulate the similarity measure of two feature trajectories.

Let us denote by $S^{(x)}$ the x-th simplified feature trajectory in a compared video sequence S and denote by $T^{(y)}$ the y-th simplified feature trajectory in a query video sequence T. Such two feature trajectories are illustrated in Fig. 3. Let us denote $S^{(x)} = \{s_1, s_2, ..., s_i, ..., s_N\}$ and $T^{(y)} = \{t_1, t_2, ..., t_i, ..., t_M\}$. The similarity of $S^{(x)}$ and $T^{(y)}$ is measured as follows.

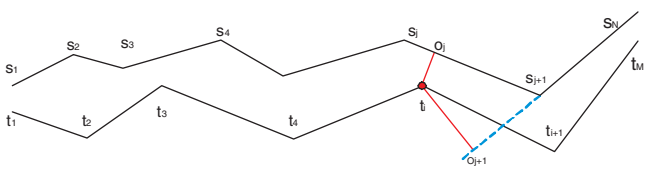

Fig. 3. Feature trajectories of two video sequences

As we know, the simplified feature trajectory $S^{(x)}$ is actually formed by $(N-1)$ ordered line segments $\overline{s_1 s_2}, ..., \overline{s_{N-1} s_N}$, denoted as $l_1^s, l_2^s, ..., l_{N-1}^s$. For each key point t_i in the simplified feature trajectory of the compared video sequence, we consider the distance from t_i to the line segment l_j^s. As shown in Fig. 3, assume that o_j is the foot of the perpendicular line from t_i to l_j^s, then o_j can be written as

$$o_j = s_j + \lambda(s_{j+1} - s_j) \tag{8}$$

where λ is a real number. Since $\overline{t_i o_j} \perp \overline{s_j s_{j+1}}$, we have

$$\overline{t_i o_j} \bullet \overline{s_j s_{j+1}} \equiv 0. \tag{9}$$

Combining Eq.(8) and Eq.(9), we obtain the expression of λ

$$\lambda = \frac{(t_i - s_j) \bullet (s_{j+1} - s_j)}{(s_{j+1} - s_j) \bullet (s_{j+1} - s_j)}, \tag{10}$$

and the distance from t_i to line segment l_j^s is composed by vertex s_j and s_{j+1}

$$d(t_i, \overline{s_j s_{j+1}}) = d(t_i, o_j) = d(t_i, s_j + \lambda * (s_{j+1} - s_j)). \tag{11}$$

However, if point o_j falls out of the range of line segment $\overline{s_j s_{j+1}}$, it is unsuitable to adopt the Eq.(11). Therefore, we define the following equation to process the out of range cases

$$d(t_i, l_j^s) = \begin{cases} d(t_i, \overline{s_j s_{j+1}}), & \text{if } 0 \le \lambda \le 1 \\ min(d(t_i, s_j), d(t_i, s_{j+1})), & \text{if } \lambda > 1 \text{ or } \lambda < 0 \end{cases} \tag{12}$$

where $d(t_i, s_j)$ and $d(t_i, s_{j+1})$ are the distances from point t_i to point s_j and to point s_{j+1}, respectively.

Based on the discussion above, we can obtain the similarity distance between two trajectories $S^{(x)}$ and $T^{(y)}$ as follow

$$dist(S^{(x)}, T^{(y)}) = \begin{cases} \frac{1}{N} \sum_{i=1}^{N} \min_{j \in [1, M-1]} d(s_i, l_j^t), & \text{if } N \le M \\ \frac{1}{M} \sum_{i=1}^{M} \min_{j \in [1, N-1]} d(t_i, l_j^s), & \text{if } N > M \end{cases} \tag{13}$$

where N and M are the number of feature points in the feature trajectories $S^{(x)}$ and $T^{(y)}$, respectively. Let us denote by $Dis(S, T)$ the dissimilarity of the video sequence S and T. From Eq.(13), we formulate the final dissimilarity function as follow

$$Dis(S,T) = \frac{1}{X+Y}\left(\sum_{x=1}^{X} \min_{y \in [1,Y]} dist(S^{(x)}, T^{(y)}) + \sum_{y=1}^{Y} \min_{x \in [1,X]} dist(T^{(y)}, S^{(x)})\right)$$

where X and Y are the number of feature trajectories in the video sequences S and T, respectively.

6 Experiments and Results

Based on our proposed framework, we implemented a compact system for video similarity detection. In our video database, we collected about 300 video clips with length ranging from 1 minute to 30 minutes. Some of them are downloaded from the Web, and some of them are sampled from the same sources with different coding formats, resolutions, and slight color modifications.

In our experiments, color histogram is extracted as the low level feature for similarity measure. Since we mainly focus on the similarity measurement method with existing features, we adopt the simple and efficient RGB color histogram as the low level feature in our experiments. Based on the extracted 64-dimension color histogram feature, we generate two kinds of signatures with different granularities. Then we conduct the performance evaluation of video similarity measures with these two kinds of signatures.

6.1 Performance Evaluation of Coarse Similarity Measure

In the coarse similarity measurement phase, we compare the performance of two kinds of PDH methods. The performance metrics used in our experiments are *average precision rate* and *average recall rate* [4]. Based on the performance metrics, we compare the performance of two kinds of PDH methods: NPDH and FPDH. The comparison result of precision-recall rate is shown in Fig. 4. We can see that the retrieval performance of FPDH is better than NPDH method. Based on the FPDH, we can obtain average 90% recall with about 50% average precision rate. This means we can filter out most of unrelated data in the coarse phase. However, we also found the average precision rate quickly drops down

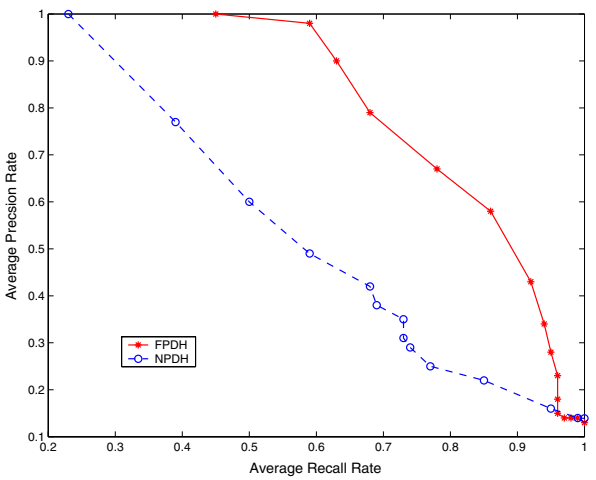

Fig. 4. Precision-recall rate comparison of NPDH and FPDH

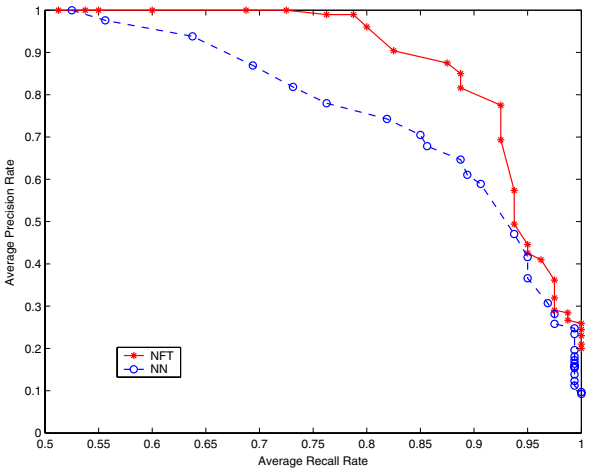

Fig. 5. Precision-recall curves comparison of NFT and NN

when the average recall rate approaches 100%. This indicates that RGB color histogram may not be an effective low level feature and more effective features need to be adopted for us to improve the performance in the future.

6.2 Performance Evaluation of Fine Similarity Measure

In order to evaluate the performance of our fine similarity measure based on the nearest feature trajectory method, we compare the retrieval performance between our NFT method and the conventional nearest neighbor (NN) method. Comparison results of these two methods are shown in Fig. 5. From the results, we can see that our proposed NFT method achieves better performance than the conventional NN method. However, we also found that even based on NFT comparison, we can, at best, achieve the best operating point at 90% precision rate with about 85% recall rate. The reason is that color feature representation is fragile to the color distortion problem. In [3], A. Hampapur et al. provide a lot of distance measure techniques using varied features, such color, shape, texture and motion, etc. We believe our proposed framework can obtain better results by using other features in the future.

7 Conclusions and Future Work

In this paper, we propose an effective two-phase framework to achieve video similarity detection. Different from the conventional way, our similarity measurement scheme is based on different granular similarity measure. In the coarse phase, we suggest the PDH technique. In the fine phase, we formulate the NFT technique. Experimental results show that our scheme is better than the conventional approach.

However, the performance of our scheme can still be improved since the color histogram based scheme is fragile to color distortion problem. In our future work, we will adopt other features to tune our video retrieval performance. We believe that better results can be achieved if we use more effective features in our framework. Also we need to enlarge our video database and test more versatile data in the future.

Acknowledgement

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4360/02E).

References

[1] S.-C. Cheung and A. Zakhor: Estimation of web video multiplicity. In *Proc. SPIEiVInternet Imaging*, vol. 3964, pp. 34-36, 2000. 373

[2] M. Naphade, M. Yeung, and B. L. Yeo: A novel scheme for fast and efficient video sequence matching using compact signatures. In *Proc. SPIE, Storage and Retrieval for Media Databases*, San Jose, CA, Jan 2000. 373, 374

[3] A. Hampapur and R. Bolle: Comparison of distance measures for video copy detection. In *Proc. of International Conference on Multimedia and Expo 2001*, 2001. 373, 374, 381

[4] S. C. Cheung and A. Zakhor: Efficient video similarity measurement and search. In *Proc. of ICIP2000*, vol. 1, pp. 85-89, Canada, Sep 2000. 373, 374, 380

[5] H. S. Chang, S. Sull, and S. U. Lee: Efficient video indexing scheme for content based retrieval. In *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1269-1279, Dec 1999. 373

[6] Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge: A framework for measuring video similarity and its application to video query by example. In *Proc. of ICIP1999*, vol. 2, pp. 106-110, 1999. 373

[7] Man-Kwan Shan and Suh-Yin Lee: Content-based video retrieval based on similarity of frame sequence. In *Proc. of International Workshop on Multi-Media Database Management Systems*, pp. 90-97, Aug 1998. 373

[8] Yi Wu, Y. Zhuang, and Y. Pan: Content-Based Video Similarity Model. In *Proc. of the 8th ACM Int. Multimedia Conf. on Multimedia*, USA, pp. 465-467, 2000.

[9] R. Mohan: Video sequence matching. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3697-3700, May 1998. 374

[10] Stefan Berchtold, Christian Böhm, and Hans-Peter Kriegel: The Pyramid-Technique: Towards Breaking the Curse of Dimensionality. In *Proc. Int. Conf. on Management of Data, ACM SIGMOD*, Seattle, Washington, 1998. 374, 375, 376

[11] S. Z. Li: Content-based Classification and Retrieval of Audio Using the Nearest Feature Line Method. In *IEEE Transactions on Speech and Audio Processing*, 2000. 377

[12] Li Zhao, W. Qi, S. Z. Li, S. Q.Yang, and H. J. Zhang: Key-frame Extraction and Shot Retrieval Using Nearest Feature Line (NFL). In *Int. Workshop on Multimedia Information Retrieval*, in conjunction with *ACM Multimedia Conf.*, Nov 2000. 377