Similarity Measurement and Detection of Video Sequences

HOI Chu-Hong

Master of Philosophy
Computer Science and Engineering

2nd Term Paper

Supervised by

Prof. LYU Rong Tsong Michael

©Department of Computer Sicence and Engineering
The Chinese University of Hong Kong

April 2003

Abstract

Efficient technique to detect the similar video sequences on the web has become one of the most important and challenging issues in multimedia and database related areas. In this paper, an original 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 technique and fine signature is based on the Nearest Feature Trajectory 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 and Hausdorff distance measure methods, our proposed similarity measurement method well incorporates the temporal order of video sequences. Experimental results show that our scheme achieves better quality results than the conventional approaches.

Contents

Chapter 1

Introduction

1.1 Motivation

Along with the rapid development of compute networks and Internet, the amount of information on the web have grown immensely in past several years. Without a central management of the web, information redundance becomes inevitable. The information redundance leads to the waste of storages and increases the difficulty of information retrieval. The situation is much more severe for multimedia content, especially for video data. Therefore, finding effective similarity measurement metric and efficient methods for video similarity detection have been proposed as an imperative issue in multimedia retrieval and web mining areas[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

1.2 Applications of Similar Video Detection

Video similarity detection has a lot of underlying applications. The first application is for copyright problems. Along with easily obtaining, editing and propagating video data on the web, more and more copyright issues are aroused today. Although watermarking technique has been proposed for the copyright issue, it can only apply to the original video content before copies

are made. Thus, it is unsuitable for the video data which have been on the web in circulation. Video copy detection is therefore proposed as a complementary approach of watermarking for the copyright issues[2].

The second useful application of video similarity detection is for information retrieval[5, 6, 7, 8]. Although current web search engines have conducted well in the text based information retrieval, they can only perform naive multimedia searching until now. Video similarity detection techniques can be integrated into current web searching engines for efficient data management and clustering of retrieval results for postprocessing purpose.

Moreover, finding similar video copies or duplications over multiple locations can provide fault tolerant services on the web[13]. While a requesting video cannot be accessed in a location due to the expired link problem, video replicas from other locations can be accessed by the request at that time. Also through finding similar video copy on the web, users can select the best accessing location with best downloading speed to facilitate their retrieval task.

1.3 Outline

In this paper, we explore effective video similarity measurement algorithms and fast similarity detection techniques. The rest of this paper is organized as follows. We first in chapter 2 cover some background knowledge of video similarity measurement and review some related solution in literature. Chapter 3 presents a two-phase similarity detection framework and briefly discusses each part of the framework. Chapter 4 discusses the coarse similarity measurement algorithm in detail and evaluates the performance. Chapter 5 describes the fine similarity measurement techniques and provides the related comparison with conventional solution. Chapter 6 gives the conclusions and our

future work.

□ **End of chapter.**

Chapter 2

Background Review

2.1 Effective Similarity Measurement

The similarity of video sequences mentioned in the paper means how large percentage of similar frames or shots shared by two video sequences. Measuring the similarity of two video sequences is similar to measure the similarity of two text documents[14]. For text documents, we compute the percentage of similar words shared by two text documents while we compute the percentage of similar frames or shots for video sequences. However, measuring the similarity of frames or shots among video sequences is more difficult to handle than text documents. To measure the similarity of frames of two video sequences, the typical approach is to represent each frame in video sequences into a high-dimensional feature vector based on a set of attributes, such as color, texture, shape and motion. Then similarity of frames or shots is computed based on a similarity metric function in the corresponding feature vectors. In past decade, a lot of research efforts are performed to find effective feature representation in image and video processing domain[2, 15, 16, 17, 18, 19]. Since the frame number of a full video sequence is usually very large, it is very time-consuming to gauge the similarity between video sequences by measuring the similarity frame by frame. In order to define the similarity measurement function, a typical

approach to gauge the similarity is based on finding the nearest key frames or key shots in two video sequences called the nearest neighbors(NN) or (k-NN) algorithm[7]. Other heuristic techniques, such as warping distance, Hausdorff distance and template matching of shot change duration can be found in [8,11,20,21,37]. In [1], we propose the nearest feature trajectory technique to perform the effective similarity measurement of video sequences. All of these methods are mainly consider to improve the precision of similarity measurement but not the efficiency problem.

2.2 Efficient Similarity Detection

Similarity measurement by sequential scanning methods is too computational complex for a large scale database. Thus, it is important to study efficient algorithms to facilitate the similarity detection. In past, there are a lot of efficient data structures have been designed to improve the similarity search in database areas[22, 23, 24, 25, 26]. Although these techniques also called Spatial Access Methods (SAM) have been widely investigated, most of them cannot scale well to high dimensional space since of the problem of "curse of dimension". In order to overcome the challenging issue, dimensionality reduction need be performed before using the SAM techniques. A typical approach is to design efficient algorithm to map the high dimensional data to a lower dimensional feature space where one of the SAM techniques can handle efficiently. The techniques are generally called the GEneric Multimedia INdexIng (GEMINI)[27].

One of the most popular feature extraction techniques is Principle Component Analysis(PCA) which is widely applied in compute vision and many other communities [28]. Multidimensional Dimension Scaling (MDS) technique is also a widely used technique to create mapping from high dimensionality to low dimen-

sional space [29]. However, these methods are too computational complex to perform the feature extraction task. In [30], the authors propose a less computational intensive technique called Fastmap algorithm to efficient mapping the high dimensional data to low dimensional feature. The complexity of Fastmap algorithm is linear with respect to the size of database. Although the heuristic algorithm is very efficient, one of its major drawback is the inefficiency problem of update operation since any update operation of Fastmap need to scan the database entirely. Other efficient technique such as Discrete Fourier Transform and Discrete Wavelet Transforms were widely explored in recent literature [31, 32]. In this paper, we propose an efficient algorithm to map the high dimensional data to low dimensional feature space [1]. The complexity of our algorithm is linear with respect to the size of the original data space.

□ **End of chapter.**

Chapter 3

A Two-Phase Similarity Detection Framework

Toward the challenging issue of fast and effective similar video detection from a large scale video database, we propose a two-phase similarity detection framework based on two kinds of signatures with different granularities, shown in Fig.1.

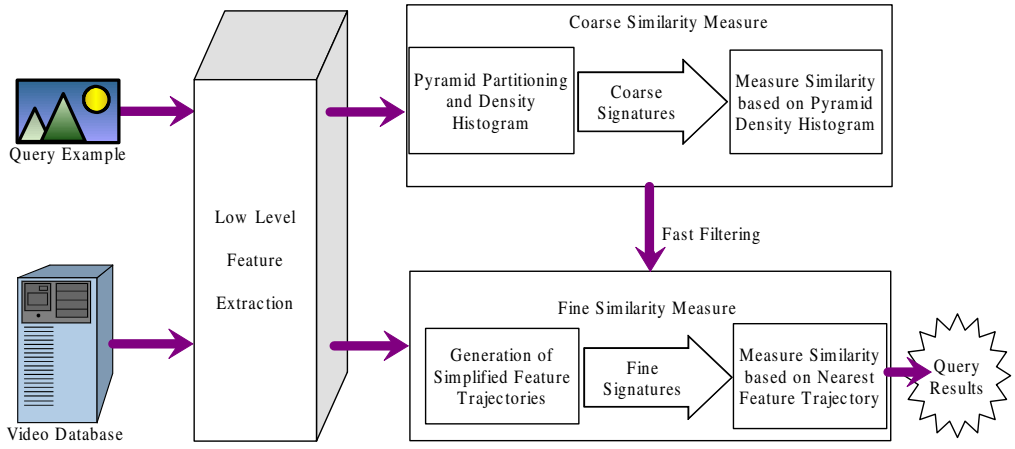


Figure 3.1: A Two-Phase Similarity Detection Framework.

In the preprocessing step, we extract the low level features of the query video example and compared video data in the video database. 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 feature

point density histograms by mapping the original data space to a new pyramid space[33], while fine signatures are obtained by generating simplified feature trajectories of video sequences. We then perform the similarity detection based on two-phase similarity measurement.

3.1 Coarse Similarity Measurement

Based on the Pyramid Density Histogram mapping technique, we map the original high dimensional data space to a low dimensional feature space. The low dimensional feature vector is called the coarse signature. Then we perform fast similarity measurement based on the coarse signature. Through the first phase, most of statistically unrelated video data are fast filtered out by coarse similarity measure based on the coarse signature.

3.2 Fine Similarity Measurement

In the second phase, fine similarity measure is performed by computing the similarity of feature trajectories of the video sequences based on the filtering results 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. Therefore, our proposed scheme can well accomplish the task of similarity detection efficiently.

□ **End of chapter.**

Chapter 4

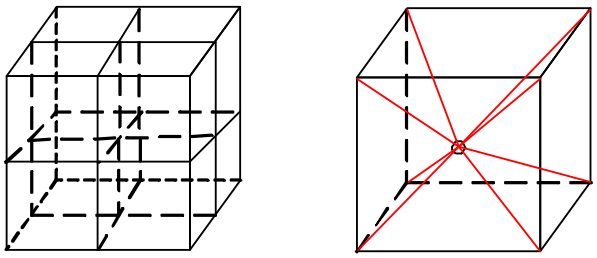
Coarse Similarity Measurement

Based on the proposed framework in previous chapter, each frame of a video sequence is considered as a feature point in the high dimension feature space after the low level feature extraction. Then a video sequence is formed by a set of feature points in a high dimension space. To approach the efficient similarity measure, it is impossible to conduct the measurement frame by frame of video sequences. In order to fast filter out the unrelated video sequences, we explore the Pyramid Density Histogram (PDH) technique as follows.

4.1 Pyramid Partitioning and Density Histogram

Pyramid partitioning technique is first proposed to solve dimension reduction and indexing problem in[12]. 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 bases[33], shown in Fig.2(b).

Suppose a video sequence S is formed by M frames corresponding to M feature points in a $d-$dimension data space.



(a) Regular Partition (b) Pyramid Partition

Figure 4.1: Partitioning of the high dimension data space

Each feature point v in the video sequence S is denoted as $v = (v_1, v_2, ..., v_d)$. Based on the pyramid partitioning technique, for a given feature point v, we assign v to the i-th pyramid by following the conditions below:

$$i = \begin{cases} j_{max}, & \text{if } (v_{j_{max}} < 0.5) \\ j_{max} + d, & \text{if } (v_{j_{max}} \geq 0.5) \end{cases} \tag{4.1}$$

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 corresponding i-th pyramid is defined as[12]:

$$h_v = |0.5 - v_{iMODd}|. \tag{4.2}$$

For each feature point v in the video sequence S, we can locate it in a unique pyramid. Through calculating the distribution density of feature points in each pyramid, we propose a pyramid density histogram technique to map the video sequence S in the original data space to the new pyramid feature space. Then each video sequence is represented as a feature vector in the new feature space, such feature vector is called the *coarsesignature* of the video sequence. In following section, we discuss three kinds of PDH technique with different extensions: *Naïve pyramid density histogram*, *Fuzzy Pyramid Density Histogram* and *General Pyramid Density Histogram.*

4.2 Naïve Pyramid Density Histogram

By directly applying the basic pyramid partitioning technique to density histogram, we obtain the original pyramid density histogram called Naïve Pyramid Density Histogram (NPDH).

Definition 4.1 (Naïve Pyramid Density Histogram) *Given a video sequence S which is formed by n feature points with d-dimension, the original data space can be mapped to a $2d$-dimension feature vector u by the Pyramid Density Histogram technique. The NPDH feature vector u denoted as $u = (u_1, u_2, ..., u_{2d})$ is calculated as: sequentially scanning each point v in sequence S, then updates the appropriate component of feature vector u by following equation:*

$$u_i = u_i + h_v \tag{4.3}$$

where i is defined in Eq.(4.1) and h_v is defined in Eq.(4.2).

By applying the NPDH mapping technique, a video sequence with N d-dimension feature points is represented as a $2d$-dimension NPDH vector.

4.3 Fuzzy Pyramid Density Histogram

From the definition of NPDH, each point v in a video sequence is totally allocated to a unique pyramid. We found that the mapping method cannot fully exploit all information in each dimension. Therefore, for each point in a video sequence, instead being completely allocated to only one pyramid, it is assigned to d pyramids in some degree with respect to the value of each dimensions. The modified technique is called Fuzzy Pyramid Density Histogram defined as follow.

Definition 4.2 (Fuzzy Pyramid Density Histogram) *Given a video sequence S which is formed by n feature points with d-dimension, the sequence in original space can be mapped to a $2d$-dimension feature vector u by the Fuzzy Pyramid Density Histogram technique. The FPDH vector u denoted as $u = (u_1, u_2, ..., u_{2d})$ is calculated as: sequentially scan each feature point v in the video sequence S, then updates the FPDH feature vector u by following equation:*

$$u_i = u_i + h_v \tag{4.4}$$

$$i = \begin{cases} j, & if\ (v_j < 0.5) \\ j + d, & if\ (v_j \geq 0.5) \end{cases} \tag{4.5}$$

where j=1,2,...,d and h_v is defined in Eq.(2).

Performance comparison result of FPDH and NPDH is shown at the end of this chapter.

4.4 General Pyramid Density Histogram

In order to obtain a general form of Pyramid Density Histogram algorithm, we extend the fuzzy pyramid density histogram to a more general algorithm called General Pyramid Density Histogram(GPDH).

Definition 4.3 (General Pyramid Density Histogram) *Given a video sequence S which is formed by N feature points with d-dimension, the sequence in original space is mapped to a $n \times d$-dimension feature vector u by the Pyramid Density Histogram technique, where n is a GPDH factor. The GPDH vector u denoted as $u = (u_1, u_2, ..., u_{nd})$ is calculated as: sequentially scan each feature point v in the video sequence S, then updates the GPDH vector u by following equation:*

$$u_i = u_i + h_v \tag{4.6}$$

in which,

$$i = (j-1) \times d + k \tag{4.7}$$

where k satisfies the inequation below,

$$\frac{k-1}{n} < v_j < \frac{k}{n}. \tag{4.8}$$

and h_v is computed as,

$$h_v = \begin{cases} |v_j - \frac{1}{n}|, & if \ \ v_j \epsilon [0, \frac{1}{n}) \\ |v_j - \frac{2k-1}{2n}|, & if \ \ v_j \epsilon (\frac{k-1}{n}, \frac{k}{n}] \\ |v_j - \frac{n-1}{n}|, & if \ \ v_j \epsilon (\frac{n-1}{n}, 1] \end{cases} \tag{4.9}$$

for all j=1,2,3,...,d.

4.5 Coarse Similarity Measure Based on PDH

Based on the proposed PDH technique, each video sequence is mapped to a *nd*-dimension feature vector as a coarse signature in the pyramid data space. We can conduct the coarse filtering based on the coarse signatures. Suppose u_q is the PDH vector for a query example and u_c is the PDH vector for a compared video sample C in a database. Let ε be the threshold of coarse similarity filtering. The coarse similarity measure is defined below.

Definition 4.4 (Coarse Similarity Measure)
Given a query video sequence Q and a compared video sequence C in database, the video sequence C is filtered out if it meets the condition below:

$$||u_q - u_c|| > \varepsilon. \tag{4.10}$$

in which, $||u_q - u_c||$ is the Euclidean distance of vector u_q and u_c, where ε is the threshold of coarse similarity measurement.

Based on the PDH technique, the similarity measurement in first phase can be very accomplished very efficiently. After the first phase, we obtain a small query result set based on a given threshold. In order to improve the precision rate, further comparison should be perform in the second phase.

4.6 Experiments and Results

Based on the proposed framework, we have 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 the coarse similarity measurement phase, we compare the performance of two kinds of pyramid density histogram methods. The performance metrics used in our experiments are *average precision rate* and *average recall rate*. Their definitions are given below. For a query example q and a given threshold δ, let $ret(q, \delta)$ denote the return set for a query under a threshold δ. Let $N(ret(q, \delta))$ denote the total number of the return set and $C(ret(q, \delta))$ denote the number of correct results in the return set. Let $M(ret(q, \delta))$ denote the number of missing correct result in the return set. Then the average precision rate is defined as,

$$Precision(\delta) = avg\frac{C(ret(q, \delta))}{N(ret(q, \delta)} \tag{4.11}$$

and the average recall rate is defined as,

$$Recall(\delta) = avg\frac{C(ret(q, \delta))}{C(ret(q, \delta) + M(ret(q, \delta))} \tag{4.12}$$

Based on the performance metrics, we compare the performance of two kinds of pyramid density histogram: NPDH and FPDH. The comparison Precision-recall rate figure is shown in Fig.(4.2). From Fig.4, we can see that the retrieval performance

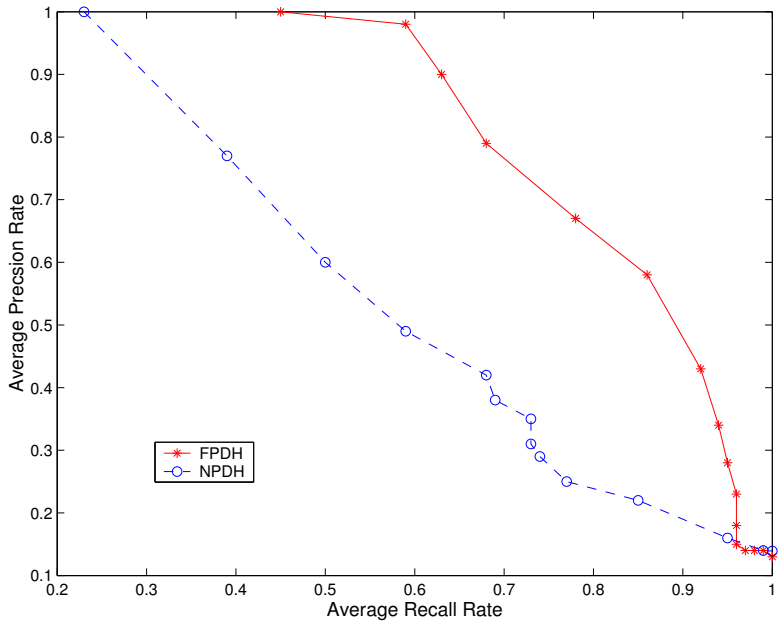


Figure 4.2: Precision-recall rate curves comparison of NPDH and FPDH.

of FPDH is better than NPDH. 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 recall rate quickly drops down when it approaches 100%. This indicates that RGB color histogram may not be an effective low level feature and we need to adopt more effective feature in our scheme to improve the recall rate in the future.

In order to evaluate the impact of the GPDH factor n, we conduct the following experiments. We adopt a series of different GPDH factors and record the retrieval performance result of each factor with a fixed recall rate 90%. The comparison result is shown in Fig.(4.3).

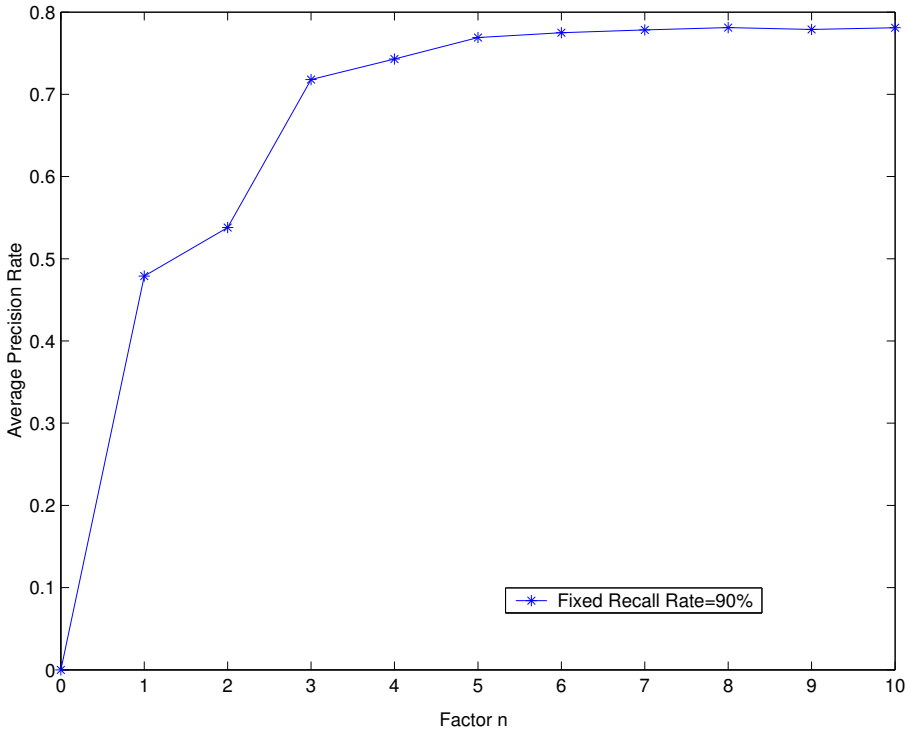


Figure 4.3: Average precision rate varies with the factor n

From the figure above, we can see that the average precision rate increases with respect to the increase of factor n. When the factor n is larger than 3, the change becomes small. The overall average precision rate is close to 80%.

□ **End of chapter.**

Chapter 5

Fine Similarity Measurement

From the previous chapter, we perform fast similarity filter in the first phase. Although we have reduced the comparison samples to a small subset through the first 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 Trajectories (NFT) technique for effective similarity measure. 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 face recognition and audio retrieval in [34][35]. It is also proved to be effective in shot retrieval of video sequence in[36]. In here, we use the similar technique to the similarity detection issue. Different from the NFL used in [36], 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 [36], we propose an 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 high dimension feature space. Two neighboring feature points form a feature line. A lot of feature lines in a shot forms a feature trajectory. A feature trajectory in a video sequence transits to another trajectory when there is a hard cut transition of shots but no gradual transition. Thus a video sequence can be represented by a series of feature trajectories called a fine signature. However, it is impractical to process the feature trajectory for all frames. Thus, we propose an efficient algorithm to generate the simplified trajectory.

Given a video sequence, we can first detect the hard cut transitions of shots. For each individual shot, we generate a simplified feature trajectory by the following descriptions. Suppose we have an individual video shot S and the number of frames in a video shot is N, denoted as $S = \{v(t_1), v(t_2), ..., v(t_N)\}$. And let S' denote the simplified feature trajectory and the number of frames in S' is N^{ψ}, denoted as $S' = \{(v'(t_1), v'(t_2), .., v'(t_{N^{\psi}}))\}$. S' is a subset of S. The optimum choice of subset S' can be obtain by following equation:

$$S' = argmin \sum_{i=1}^{N} ||v(t_i) - v'(t_i)|| \tag{5.1}$$

$$v'(t) = \sum_{j=1}^{N^{\psi}} l_j(t) \tag{5.2}$$

$$l_j(t) = v'(t_j) + \frac{v'(t_{j+1}) - v'(t_j)}{t_{j+1} - t_j} \tag{5.3}$$

However it is time-consuming to obtain the best answer of Eq.(5.1) with global minimum error. Thus we propose the following alternative algorithm which is effective and efficient to achieve a local optimum answer.

We assume that a frame is less important if it is more similar to its two neighbor frames since it can be well estimated by its neighbors. Thus we can reduce less important points one by one according to measuring the local similarity of the trajectory[18]. After filtering out the less important points, the remaining N^{ψ} points should still represent the global shape of the feature trajectory.

We formally assume that $v_k\{k = 1, 2, ..., N\}$ represent frames in a video sequence. We define a local similarity measure function $LR(v_k)$, which denotes the similarity between its neighbors. Although the curvature at point v_k is an intuitional measure function for a curve, it is unreasonable to compute the curvature since the curve is formed piecewise by line segments which are not smooth. Thus we define a similarity measure function as:

$$LR(v_k) = |d(v_k, v_{k-1}) + d(v_{k+1}, v_k) - d(v_{k+1}, v_{k-1})| \tag{5.4}$$

where $d(v_i, v_j)$ means the distance between point i and point j. Obviously, v_{k-1},v_k and v_{k+1} satisfy the triangle-inequality relation. In the special case, if $LR(v_k) = 0$, 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 is N^{ψ}.

5.2 Similarity Measure Based on the Nearest Feature Trajectory

Based on the fine signatures discussed above, we proposed a fine similarity measure of video sequences. Given two video sequences, the similarity 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.

Suppose the x-th simplified feature trajectory in a compared video sequence S is denoted as $S^{(x)}$ and the y-th simplified feature trajectory in a query video sequence T is denoted as $T^{(y)}$. Such two feature trajectories are illustrated in Fig.2. Let

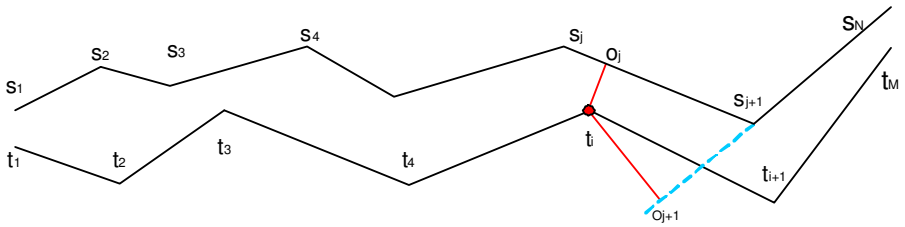


Figure 5.1: Feature trajectories of two video sequences

$S^{(x)} = \{s_1, s_2, ..., s_i, ..., s_N\}$ and $T^{(y)} = \{t_1, t_2, ..., t_i, ..., t_M\}$, then we define $Dis(S^{(x)}, T^{(y)})$ as the dissimilarity measure function of two feature trajectories. Since slide-window based subsequence pattern matching method overemphasizes the order of sequence, it is not suitable here to handle two video sequences with different frame rates. Therefore, we propose a point-to-line similarity measurement method as follows.

As we know, the simplified feature trajectory $S^{(x)}$ is actually formed by $(N-1)$ ordered line segments $\overline{s_1 s_2}, ..., \overline{s_{N-1} s_N}$, de-

noted 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.2, 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:

$$o_j = s_j + \lambda(s_{j+1} - s_j) \tag{5.5}$$

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

$$\overline{t_i o_j} \bullet \overline{s_j s_{j+1}} \equiv 0. \tag{5.6}$$

Combining Eq.(5.5) and Eq.(5.6), we obtain the expression of λ

$$\lambda = \frac{(t_i - s_j) \bullet (s_{j+1} - s_j)}{(s_{j+1} - s_j) \bullet (s_{j+1} - s_j)}, \tag{5.7}$$

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

$$d(t_i, \overline{s_j s_{j+1}}) = d(t_i, o_j) = d(t_i, s_j + \lambda * (s_{j+1} - s_j)). \tag{5.8}$$

From Eq.(5.8), we know that the distance from point t_i to line l_j^s is equal to the distance from point t_i and the foot of the perpendicular point o_j, where o_j is within the range of line segment$\overline{s_{j-1} s_j}$ or in its extension. However, if point o_j falls in the extension part of line $\overline{s_j s_{j+1}}$, it is unsuitable from our discussion. Obviously, when $\lambda < 0$ or $\lambda > 1$, o_j falls out of the range of line segment $\overline{s_j s_{j+1}}$; otherwise o_j falls in the range of line segment $\overline{s_j s_{j+1}}$. In the special case, o_j is equal to s_j when $\lambda = 0$ and o_j is equal to s_{j+1} when $\lambda = 1$. In order to minimize the error caused by the out-of-range cases, we can define the distance from point t_i to line segment l_j^s as

$$d(t_i, l_j^s) = \begin{cases} d(t_i, \overline{s_j s_{j+1}}), \text{ if } 0 \leq \lambda \leq 1 \\ \\ min(d(t_i, s_j), d(t_i, s_{j+1})), \text{ if } \lambda > 1 \text{ or } \lambda < 0 \end{cases} \tag{5.9}$$

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 follows:

$$dist(S^{(x)}, T^{(y)}) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} min_{j\in[1,M-1]} d(s_i, l_j^t), & \text{if } N \leq M \\ \\ \frac{1}{M}\sum_{i=1}^{M} min_{j\in[1,N-1]} d(t_i, l_j^s), & \text{if } N > M \end{cases} \tag{5.10}$$

where N and M are the number of feature points in the feature trajectories $S^{(x)}$ and $T^{(y)}$, respectively. From Eq.(5.10), we can define the final dissimilarity measure function between two video sequences S and T as follows:

$$\begin{aligned} Dis(S,T) = \frac{1}{X+Y}\Big(&\sum_{x=1}^{X} min_{y\in[1,Y]} dist(S^{(x)}, T^{(y)}) \\ &+ \sum_{y=1}^{Y} min_{x\in[1,X]} dist(T^{(y)}, S^{(x)})\Big) \end{aligned} \tag{5.11}$$

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

5.3 Experiments and Results

In order to evaluate the performance of our fine similarity measure based on the nearest feature trajectory (NFT) method, we compare the retrieval performance between our NFT method and the conventional nearest neighbor (NN) and Hausdorff distance measure method [37]. The comparison results of these two methods are shown in Fig.5. From the experimental results shown in Fig.(5.1), we can see that our proposed NFT method achieves better performance than the conventional NN and Hausdorff method. However, we also found that even based on NFT comparison, we can, at best, achieve the best operating

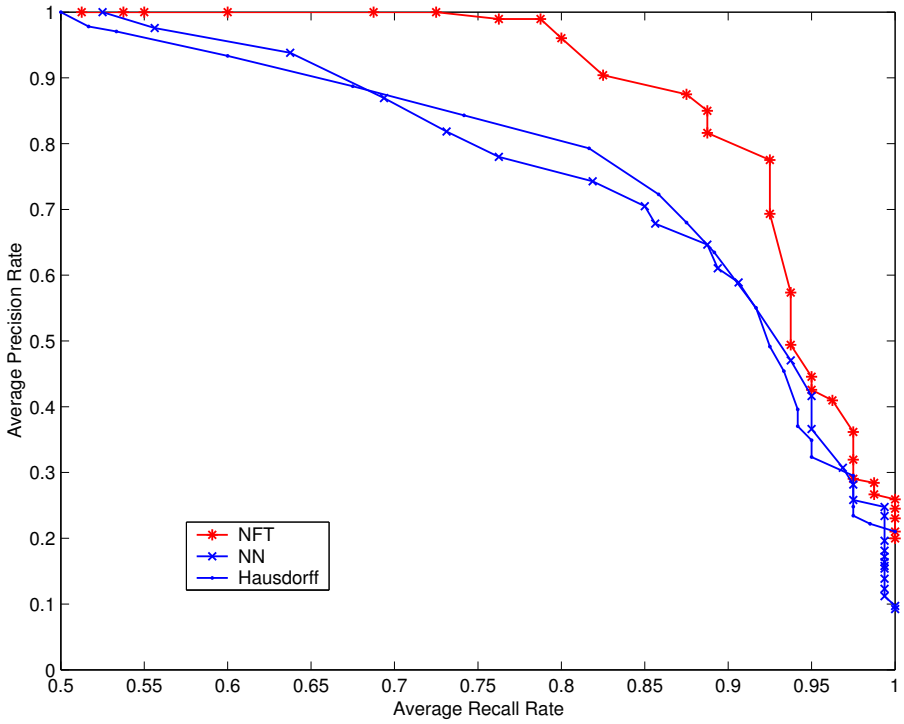


Figure 5.2: Precision-Recall curves comparison of two methods.

point at 90% precision rate and 85% recall rate. The reason is that color feature representation is fragile for the color distortion problem. In [2], A. Hampapur et al. provide the comparison of a lot of distance measures with different attributes such as color, shape, texture and motion, etc. We believe our proposed framework can obtain better results by combining other feature representation such as shape and motion feature in the future.

□ **End of chapter.**

Chapter 6

Conclusion 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 Pyramid Density Histogram technique. In the fine phase, we formulate the Nearest Feature Trajectory technique. Experimental results show that our scheme is better than the conventional approach.

However, the result of our scheme can still be improved since the color histogram based scheme is fragile for color distortion problem. In our future work, we will adopt other features to tune the 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.

□ **End of chapter.**

Bibliography

[1] Chu-Hong Hoi, Wei Wang, and Michael R. Lyu, "A novel scheme for video similarity detection," To appear in *International Conference of Image and Video Retrieval (CIVR2003)*, Urbana, IL, USA, 2003.

[2] A. Hampapur and R. M. Bolle, "Comparison of distance measures for video copy detection," In *Proc. of Int. Conf. on Multimedia and Expo*, Aug. 2001.

[3] 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*, Volume 3972, pages 564-572, San Jose, CA, Jan 2000.

[4] S. Cheung and A. Zakhor, "Efficient video similarity measurement and search," In *Proc. of International Conference on Image Processing*, vol.I, pp. 85-89, British Columbia, Canada. September 10-13, 2000.

[5] Yap-Peng 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 International Conference on Image Processing*, Volume:2, Page(s):106-110, 1999.

[6] 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, Page(s):90 -97, 5-7 Aug, 1998.

[7] Yi Wu, Yueting Zhuang and Yunhe Pan, "Content-Based Video Similarity Model," In *Proc. of the 8th ACM International Multimedia Conf. on Multimedia*, Marina del Rey, CA, USA, pp.465-467, October 30-November 04, 2000.

[8] M.R. Naphade, R. Wang, and T.S. Huang, "Multimodal pattern matching for audio-visual query and retrieval," in Proceedings of the Storage and Retrieval for Media Databases 2001, San Jose, USA, jan 2001, vol. 4315, pp. 188-195, 2001.

[9] R. Mohan, "Video sequence matching," In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing 1998*, Volume:6, Page(s):3697-3700, 12-15 May, 1998.

[10] S. Cheung. "Efficient Video Similarity Measurement and Search," Ph.D. Dissertation of UC-Berkeley. Committee members: Avideh Zakhor (chair) , Larry Rowe, Ray Larson, December 2002.

[11] R. Lienhart, W. Effelsberg, and R. Jain, VisualGREP, "A systematic method to compare and retrieve video sequences," in Proceedings of storage and retrieval for image and video databases VI. SPIE, vol. 3312, pp. 271-82, Jan. 1998.

[12] 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. 1269-79, Dec 1999.

[13] S. Cheung and A. Zakhor, "Estimation of Web Video Multiplicity," In Proceedings of the SPIE - Internet Imaging,

vol. 3964, pp. 34-46. San Jose, California. January 22-28, 2000.

[14] N. Shivakumar and H. Garcia-Molina,"Finding near-replicas of documents on the web," in *World Wide Web and Databases. International Workshop WebDB'98*, Valencia, Spain, pp. 204-12, Mar 1998.

[15] Yu-Fei Ma, Hong-Jiang Zhang, "Motion Texture: A New Motion based Video Representation," In Proceeding of 2002 International Conference on Pattern Recognition, ICPR, August, 2002.

[16] V. Castelli and L. D. Bergman, Eds., "Image Databases: Search and Retrieval of Digital Imagery," John Wiley & Sons, 2001.

[17] D. Bhat and S. Nayar, "Ordinal measures for image correspondence," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 Issue: 4, pp. 415V423, April 1998.

[18] Wei Wang and M.R. Lyu, "Automatic Generation of Dubbing Video Slides for Mobile Wireless Environment," In *Proc. of IEEE International Conf. on Multimedia and Expo*, Orlando, Florida, July 27-30, 2003.

[19] A. Hampapur and R. M. Bolle, "Feature based indexing for media tracking," in *Proc. of Int. Conf. on Multimedia and Expo*, pp. 67-70, Aug. 2000.

[20] D. Adjeroh, I. King, and M.C. Lee, "A distance measure for video sequence similarity matching," in Proceedings International Workshop on Multi-Media Database Management Systems, Dayton, OH, USA,pp. 72-9, Aug. 1998.

[21] S. H. Kim and R.-H. Park, "An efficient video sequence matching using the Cauchy function and the modified Haus-

dorff distance," in Proc. SPIE Storage and Retrieval for Media Databases 2002, pp. 232-239, San Jose, CA, USA, Jan. 2002

[22] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece. 1997, pp. 426-435,1997.

[23] R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in Proceedings of the 24th International Conference on Very-Large Databases (VLDB'98), pp. 194205, (New York, NY, USA), August 1998. P. Indyk, G. Iyengar, and N.

[24] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in Proceedings of the 25th International Conference on Very-Large Databases (VLDB'99), (Edinburgh, cotland), 1999.

[25] E. Kushilevitz, R. Ostrovsky, and Y. Rabani, Efficient search for approximate nearest neighbor in high dimensional spaces," in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, pp. 61423, May 1998.

[26] Guttman, A., "R-trees: A dynamic index structure for spatial searching," In Proc. ACM SIGMOD Conf., pp 47-57, 1984

[27] Faloutsos, C., Ranganathan, M., and Manolopoulos, Y., "Fast subsequence matching in time-series databases," In Proc. ACM SIGMOD Conf., Minneapolis, 1994.

[28] Hotelling H., "Analysis of a complex of statistical variables into principal components," J. Educ. Psych., 24:417C441, 498C520, 1933.

[29] M. Beatty and B.S. Manjunath, "Dimensionality reduction using multi-dimensional scaling for content-based retrieval," Image Processing, 1997. Proceedings., International Conference on , 26-29 Oct 1997 Page(s): 835 -838 vol.2, 1997.

[30] C. Faloutsos and King-Ip Lin, "Fastmap: a fast algorithm for indexing, datamining and visualization of traditional and multimedia datasets," in Proceedings of ACM-SIGMOD,pp. 163-174, May 1995.

[31] D. Rafiei and A. Mendelzon, "Efficient retrieval of similar time sequences using DFT," In Proc. of the FODO Conf., Kobe, Japan, November 1998.

[32] Chakrabarti, M. N. Garofalakis, R. Rastogi, and K. Shim, "Approximate query processing using wavelets," In The VLDB Journal, pages 111C122, 2000.

[33] 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.

[34] S. Z. Li and J. Lu: Face recognition based on nearest linear combinations, in *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp.439-443, March 1999.

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

[36] Li Zhao, Wei Qi, S.Z.Li, S.Q. Yang, H.J. Zhang, "A New Content-based Shot Retrieval Approach: Key-frame Extraction based Nearest Feature Line (NFL) Classification," ACM Multimedia Information Retrieval 2000, Los Angeles, USA, Oct 2000.

[37] S. H. Kim and R.-H. Park, "An efficient algorithm for video sequence matching using the Hausdorff distance and the directed divergence," in *Proc. SPIE Visual Communications and Image Processing 2001*, vol. 4310, pp. 754-761, San Jose, CA, Jan. 2001.