Trajectory-based Video Retrieval by String Matching

JunWei Hsieh*, Shang-Li Yu, and Yung-Sheng Chen
Department of Electrical Engineering, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li 320, Taiwan
shieh@saturn.yzu.edu.tw

ABSTRACT

This paper proposes a trajectory-based video retrieval system to retrieve desired videos from a video database through string matching. First, in order to represent each trajectory, a hybrid technique is proposed for representing its semantic meanings and geometrical properties. At this scheme, we use the Bezier basis functions to interpolate some lost control points of each represented trajectory. Then, the distance between any two trajectories can be measured by comparing the positions of sampling points extracted along their Bezier approximations. In addition, this hybrid method uses a novel labeling technique for converting a trajectory into a string. This string representation can give more semantic information in interpreting a trajectory and make important improvements in video classification. More importantly, the problem of partial matching will become easy and can be efficiently solved by a string matching technique. Experimental results have proved the superiority of our proposed method.

1. Introduction

When browsing a video, moving objects usually attract most of users' attentions. People will be more interested in the actions of a car or an actor than backgrounds in the video. Compared with still features, motion features can offer better temporal information for a video retrieval system to easily and quickly find desired video clips. Therefore, there have been a number of researchers [1]-[5] who have devoted themselves to investigating motion characteristics on video retrieval. Prior techniques in video retrieval via motions can be divided two kinds of methods, i.e., the camera-based and the object-based ones. For the first one, camera motions can be categorized to several types like: zooming in or out, panning left or right, tilting up or down, and so on. However, the camera-based method is ineffective in video retrieval since many irrelevant video contents have similar camera motions and the attentions of users are mainly paid to moving objects rather than cameras. For the object-based method, two kinds of modeling techniques are used to capture motion characteristics, i.e., the one for modeling objects' relationships and the other for modeling object's motion trajectories. For the first one, Bimbo et al. [2] used several symbolic representations to describe the relationships between objects for video retrieval. Arndt and Chang [3] used 2D strings and the set theory to represent object spatial relations and changes. Although symbolic representations can represent objects' relationships in temporal domain very well, lots of time and efforts are needed to label each object and its positions. For the trajectory modeling, VideoQ [1] provides good example of video searching based on a set of visual features like color, texture, shape, and motions. In addition, Dagtas *et al*.[5] proposed a trajectory-based model and a trail-based model for indexing videos by taking advantages of the Fourier transform and Mellin transform. However, the extension from 1D trajectory to 2D image will complicate the complexity of motion matching. All these trajectory modeling methods lack of capabilities to put semantic interpretations on each trajectory. In addition, these methods are weak in partial matching for retrieving desired trajectories from the video database if a partial trajectory is given.

In this paper, we propose a trajectory-based video retrieval system which can analyze different moving trajectories and then retrieve related video clips through string matching. For representing each trajectory, a hybrid method is proposed for capturing its semantic meanings and geometrical properties. The hybrid method includes two complementary schemes for trajectory representation, i.e., the string-based and the sketch-based ones. The string-based method can automatically convert a trajectory to a string and match trajectories according to their semantic meanings. The sketch-based method projects a trajectory on a set of Bezier functions and matches trajectories based on their moving positions. The string-based method can easily tackle the most difficult "partial matching" problem of video retrieval at a semantic level and the sketch-based one can provide good invariance properties in dealing with the spatial shift and scaling invariant retrieval. When browsing, the string-based method is first used to pre-cluster trajectories according to their semantic meanings and then the sketch-based method is used to finely classify the remained trajectories based on their trajectory positions. Thus, the proposed scheme can access desired video clips more accurately and effectively than traditional trajectory-based methods. Experimental results have proved that the proposed method indeed achieves great improvements in terms of indexing accuracy, robustness, and stability.

2. Sketch-based Trajectory Indexing

Fig. 1 shows the flowchart of the proposed scheme. First of all, we use a sketch-based representation method to represent a trajectory according to its corresponding moving positions. The scheme can capture its

0-7803-8554-3/04/$20.00 ©2004 IEEE.

geometrical features at lower level. For capturing the semantic meanings of this trajectory, a string-based method is then proposed to understand a trajectory at a middle level. Then, through integration, the hybrid scheme can retrieve desired video clips extremely accurately. In what follows, the sketch-based method is first described and then details of the string-based scheme are discussed in Section 3.

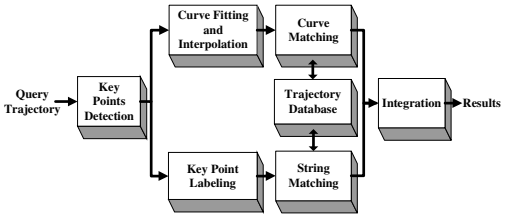


Fig. 1 Details of our proposed motion-based video retrieval system.

2.1 Key Point Selection

Assume that T is a trajectory with n_T points, i.e., $T=\{p_0, p_1, ..., p_{n_T-1}\}$. Before matching, we should first use a sampling method to detect all high curvature points from the T as key points for trajectory representation. Let p be a point in the T. Its angle can be calculated by this equation:

$$\alpha(p)=\cos^{-1}\frac{\| p-p^+\|^2+\| p-p^-\|^2-\| p^--p^+\|^2}{2\| p-p^-\|\times\| p-p^+\|},$$

where p^+ and p^- are two points specified from both sides of p. Then, according to this angle, if α is larger than a threshold, i.e., 150°, the point p is selected as a candidate of key points.

2.2 Key Point Labeling and Segment Extraction

After extracting all key points from the T, for segmenting the T to several segments and converting it to a string, in what follows, a labeling technique will be proposed to label each key point. Assume that p is one key point in T with two neighbors q_p^- and q_p^+ which are selected from both sides of p. With these two points, the direction of p can be decided according to the cross product of $\overrightarrow{q_p^-p}$ and $\overrightarrow{q_p^-q_p^+}$. If the value is positive, p is clockwise; otherwise, it is anticlockwise. Thus, the p can be labeled according the following rule:

$$Label(p)=\begin{cases}'N', \text{ if } \alpha(p)\ge 5^\circ \text{ and } dir(p) \text{ is clockwise},\\ 'C', \text{ if } \alpha(p)<5^\circ,\\ 'P', \text{ otherwise},\end{cases} \quad (1)$$

where $\alpha(p)$ is the angle of p and the 'C' denotes a sharp angle change of the p.

Through the labeling, if two key points have different labeling symbols, a segment breaking will exist in them. The labeling change will provide cues for segmenting the T to different segments. The segments can provide helpful information for partial matching.

2.3 Curve Fitting

Given two trajectories, their dissimilarity can be measured directly on the position differences between their sampling points. However, for reducing the size of database, each trajectory should be fitted to a motion model. The modeling process can gain not only the advantage of the reduction of database size but also the recovering of all missed points in a trajectory. In this paper, the Bezier approximation is used for specifying the interpolations between key points. Given a trajectory segment T_i with its set CP_{T_i} of control points, the Bezier curve of T_i can be represented as:

$$B_{T_i}(t)=\sum_{j=0}^{m_i} p_j^i J_{m_i,j}(t),$$

where $J_{n,i}(t)$ is the blending function defined as $C_i^n t^i(1-t)^{n-i}$ and $C_i^n=n!/[i!(n-i)!]$, m_i the number of elements in $|CP_{T_i}|$, and $p_j^i\in CP_{T_i}$. Assume there are N_{T_i} sampling points from T_i with a sampling period Δ_{T_i}. Then, the discrete form $T_i^B(k)$ of the $B_{T_i}(t)$ can be represented:

$$T_i^B(k)=\sum_{j=0}^{m_i} q_j J_{m_i,j}(k\Delta_{T_i}), \quad (2)$$

where $k=0,1,2,...,N_{T_i}-1$.

2.4 Normalization for Shift-and-Scale Invariance and Dissimilarity Measurement

After fitting, for keeping the shift-and-scaling invariance of the trajectory T, a normalization process should be applied to T before matching. First of all, for keeping the shift invariance, all data in T are shifted by setting the head point of the T as the original, i.e.,

$$T(k)=T(k)-T(0) \quad \text{for } k=0,\ldots,N-1. \quad (3)$$

Let $MaxX_T$, $MaxY_T$, $MinX_T$, and $MinY_T$ be the maximum and minimum x and y coordinates of $\{T(k)\}_{k=0,...,N-1}$, respectively. The scaling invariance can be maintained by normalizing all the points in $\{T(k)\}_{k=0,...,N-1}$ as follows:

$$\tilde{x}_{T(k)}=x_{T(k)}/\Delta x_T \quad \text{and} \quad \tilde{y}_{T(k)}=y_{T(k)}/\Delta y_T, \quad (4)$$

where $\Delta x_T=MaxX_T-MinX_T+1$ and $\Delta y_{T_U}=MaxY_T-MinY_T+1$. In addition to trajectory position, the speed of the T after normalization can be obtained by differential operator:

$$\tilde{v}^x_{T(k)}=\frac{\tilde{x}_{T(k+1)}-\tilde{x}_{T(k-1)}}{2} \quad \text{and} \quad \tilde{v}^y_{T(k)}=\frac{\tilde{y}_{T(k+1)}-\tilde{y}_{T(k-1)}}{2}. \quad (5)$$

Assume that $\{Q(k)\}_{k=0,...,N-1}$ and $\{D(k)\}_{k=0,...,N-1}$ are the discrete forms of a query trajectory $Q(t)$ and a target trajectory $D(t)$ in the database, respectively. Then, their position and speed dissimilarities can be obtained by calculating their Euclidean distances,

i.e., $d^{trj}(Q,D)$ and $d^{s}(Q,D)$. Then, the visual distance between $Q(t)$ and $D(t)$ can be defined as:

$$\varepsilon(Q,D) = w_{trj} d^{trj}(Q,D) + w_{speed} d^{s}(Q,D). \quad (6)$$

The values of w_{trj} and w_{speed} are equally set, i.e., 0.5.

3. String-based Trajectory Indexing

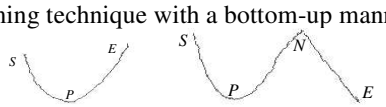
In Section 2, we have proposed a sketch-based scheme to describe trajectories and measure their dissimilarities according to their positions and speeds. When complex or partial trajectories are given, the scheme will lose its accuracy and flexibility in solving the problem of partial matching. Thus, in what follows, a string-based scheme is proposed to represent a trajectory with a set of symbol. This scheme can make a pre-classification for classifying complex trajectories. Thus, many impossible video clips can be filtered out in advance. Therefore, significant improvements in retrieval accuracy and flexibility can be achieved.

3.1 Classification through String Matching

According to the labeling algorithm described in Section 2.2, each trajectory can be converted to a string. For example, Fig. 2 shows two kinds of trajectories whose string representations are "SPE" and "SPNE", respectively. These two trajectories are different but have some commonality. Assume that Q and D are two trajectories with the corresponding strings S_Q and S_D, respectively. Their dissimilarity can be easily measured by calculating the edit distance between S_Q and S_D. The edit distance is the minimum number of edit operations required to change S_Q into S_D. The operations include replacements, insertions, and deletions. Let $C^I_{i,j}$, $C^R_{i,j}$ and $C^D_{i,j}$ be the costs of the operations "replacement", "insertion", and "deletion" performed in the ith and jth characters of S_Q and S_D, respectively. Then, the edit distance $D^e_S(i,j)$ between strings $S_Q[0..i]$ and $S_D[0..j]$ can be rewritten as

$$D^e_S(i,j) = \min[D^e_S(i-1,j) + C^D_{i,j},\ D^e_S(i,j-1) + C^I_{i,j},\ D^e_S(i-1,j-1) + C^R_{i,j}]. \quad (7)$$

In this equation, the costs $C^I_{i,j}$, $C^R_{i,j}$ and $C^D_{i,j}$ are set to $\rho + (1-\rho)\alpha(i-1,j)$, $\rho + (1-\rho)\ \alpha(i,j-1)$, and $\alpha(i-1,j-1)$, respectively, where ρ is less than 1 and set to 0.1 in this paper. $\alpha(i,j)$ is a function which is 0 if $S_Q(i) = S_D(j)$ and 1 if $S_Q(i) \neq S_D(j)$. This equation can be efficiently calculated by a dynamic programming technique with a bottom-up manner.



(a) (b)

Fig. 2: Two types of trajectories with different string representation.

In addition to the edit distance between S_Q and S_D, we also want to know the associated optimal edit transcript. The information can provide important cues in finding optimal sub-trajectories from video databases. The optimal edit transcript can be well found by using a matrix $E_{S_Q,S_D}(i,j)$ to record all associated edit operations when calculating $D^e_S(i,j)$. In what follows, details of string matching for calculating the edit distance between S_Q and S_D are described. When calculating, the matrix $E_{S_Q,S_D}(i,j)$ is also constructed.

Algorithm for String Matching

S1: Initialize the matrix $D^e_S(i,j)$:

1.1: $D^e_S(i,0) = i + \alpha(0,0)$ for all $i < l_Q$ and

$D^e_S(0,j) = j + \alpha(0,0)$ for all $j < l_D$.

S2: For i=1 to l_Q -1 do

For j=1 to l_D -1 do

2.1: Update $D^e_S(i,j)$ according to Eq. (7).

2.2: If $D^e_S(i,j) = D^e_S(i-1,j-1) + 1$, $E_S(i,j)$ = 'R'.

2.3:If $D^e_S(i,j) = D^e_S(i-1,j-1)$, $E_S(i,j)$ ='-'.

2.4: If $D^e_S(i,j) = D^e_S(i,j-1) + \rho$, $E_S(i,j)$ ='I'.

2.5: If $D^e_S(i,j) = D^e_S(i-1,j) + \rho$, $E_S(i,j)$ = 'D'.

S3: Return $D^e_{S_Q,S_D}(l_Q - 1, l_D - 1)$ and the $E_S(i,j)$.

After string matching, the optimal edit transcript can be obtained by back tracing the matrix E_{S_Q,S_D}.

3.2 Scheme Integration

In this section, an integration scheme will be proposed to build a flexible, scalable, and accurate system for motion-based video retrieval. Given two trajectories $Q(t)$ and $D(t)$, their edit distance $e\varepsilon(Q(t),D(t))$ can be calculated by the above string matching. In addition, with the matrix E_{S_Q,S_D}, the longest common substrings Γ_{S_Q} and Γ_{S_D} between $Q(t)$ and $D(t)$ can also be found. Assume that $Q_{U^i_q}$ and $D_{U^j_d}$ are two elements in Γ_{S_Q} and Γ_{S_D}, respectively. Then, the visual distance $\varepsilon(D_{U^i_q}, Q_{U^j_d})$ between $Q_{U^i_q}$ and $D_{U^j_d}$ can be calculated by Eq.(6). Then, the visual distance between $Q(t)$ and $D(t)$ can be calculated as:

$$v\varepsilon(Q(t),D(t)) = \min_{Q_{U^i_q} \in \Gamma_{S_Q}} w_{Q_{U^i_q}} \times \min_{D_{U^j_d} \in \Gamma_{S_D}} \varepsilon(Q_{U^i_q}, D_{U^j_d}), \quad (8)$$

where $w_{U^i_q} = (1 + \| Q(t) \|^2) \Big/ (1 + \| Q_{U^i_q} \|^2)$. Based on the visual distance and edit distance, their integrated distance between $Q(t)$ and $D(t)$ can be defined as:

$$Err(Q(t),D(t)) = e\varepsilon(Q(t),D(t)) \times v\varepsilon(Q(t),D(t)). \quad (9)$$

Then, each trajectory $D^n(t)$ in the database can be well sorted and indexed according to Eq.(9).

4. Experimental Results

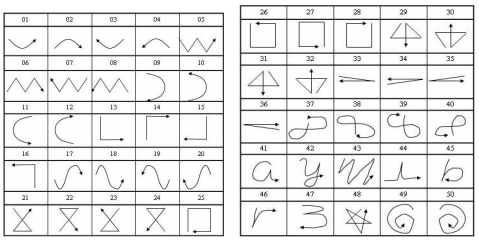


Fig. 3 Trajectory database

In order to analyze the performance of our approach, a database containing 2500 trajectories which come from 50 categories is generated. Fig. 3 shows all types of motion trajectories used in our database. In addition, 200 real video clips are also used to examine the performance of our video retrieval system. In order to make fair comparisons with other methods, several methods were implemented including the Fourier descriptor, the scheme described in MPEG 7 [4][6], the direct trajectory matching described in [5], and our proposed method. Fig. 4 shows the comparison between our method and the scheme proposed in MPEG 7 when a "*W*" type of motion trajectory is input. The accuracies of both schemes are 16% and 80%, respectively. If the results of partial matching are considered correctly, the accuracy of our scheme will be 98%. Since the string matching method can filter out many impossible trajectories in advance, significant improvement of accuracy is achieved by our scheme.

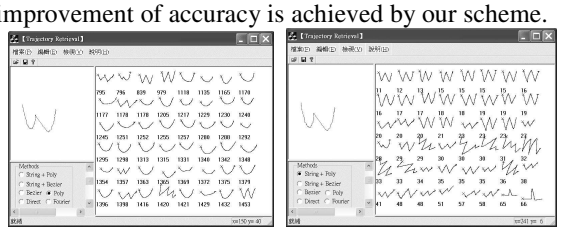


Fig. 4: Results of retrieving the "*W*" type of motion trajectory.

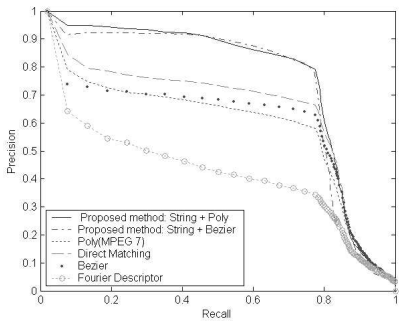


Fig.5: Performance comparisons among different approaches when the recall-precision curve is used.

Fig.5 shows the comparison results of these methods when the precision-recall graph is used. The method of Fourier descriptors performs the worst. The scheme of MPEG7 denoted by "Poly" is comparable to the one of "Bezier". For the direct matching scheme, since it puts more efforts to solving the alignment problem, more accurate retrieval results can be achieved than above three methods. However, our method sill performs the best since the usage of string matching can filter out most impossible candidates in advance. Due to the filtering capability, the proposed method has quite improvements in retrieval accuracy, nearly 20%. The average of retrieval time in our proposed method is less than 0.15 seconds under the experimental database.

For the comparison of real sets of video clips, we adopted the similar comparison method described in [5]. The average accuracies of each method are shown in Table 1. Clearly, our method much outperforms than other methods.

String + Poly	String + Bezier	Direct Matching	Poly	Bezier	Fourier
95.2%	94.3%	88.2%	86.6%	85.4%	65.3%

Table 1: Average accuracies of different methods.

6. Conclusions

In this paper, a novel string-based indexing scheme was proposed for indexing different interested videos from the database. This method has good capabilities in partial matching and approximated matching. Even though partial trajectories are handled, desired video clips still can be very accurately retrieved.

Acknowledges

This work was supported in part by National Science Council of Taiwan under Grants NSC92-2213 -E-150-049.

References

[1] S. F. Chang, et al., "A Fully Automated Content-based Video Search Engine Supporting Spatiotemporal Queries," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 602-615, Sep. 1998.

[2] A. D. Bimbo, et al., "Symbolic Description and Visual Querying of Image Sequences Using Spatiotemporal Logic," *IEEE Trans. Knowledge and Data Engineering*, vol. 7, pp. 609-622, Aug. 1995.

[3] T. Arndt and S. K. Chang, "Image Sequence Compression by Iconic Indexing," *Proc. IEEE VL' 89 Workshop on Visual Language*, Roma Italy, pp.177-182, Sep. 1989.

[4] S. Jeannin and A. Divakaran, "Mpeg-7 Visual Motion Descriptors," IEEE Trans. Circuit System Video Tech., vol. 11, No. 6, pp. 720-724, Jun1 2001.

[5] S. Dagtas et al., "Models for Motion-Based Video Indexing and Retrieval," IEEE Trans. Image Processing, vol. 9, no. 1, pp. 88-101, Jan. 2000.

[6] W. Chen and S.-F. Chang, "Motion Trajectory Matching of Video Objects," *SPIE/IS&T Storage and Retrieval for Media Databases,* San Jose CA, January 2000.