Video Retrieval Based on Video Motion Tracks of Moving Objects

Pei-Yi Chen and Arbee L.P. Chen*

Department of Computer Science
National Tsing Hua University, Hsinchu, Taiwan 300, R.O.C

ABSTRACT

Motion track is an important feature to show the spatio-temporal relationship of a video object in a video. In this paper, we propose a novel motion track representation based on MPEG-7 motion descriptor. A new descriptor is proposed to represent the motion track in the X-Y plane and the trend of velocity changes. Moreover, a new similarity measure for comparing two motion tracks based on the motion trajectory and velocity differences is proposed. The trajectory is compared by the properties of the polynomials, and the velocity is compared by the different trends. Furthermore, the motion track segmentation method is proposed to handle a complicated motion behavior and the relevance feedback is used to improve the query results. Experiment results show that this approach has a higher precision than existing approaches.

Keywords: motion track, MPEG-7, similarity measure, motion track segmentation, relevance feedback

1. INTRODUCTION

A *motion track* is a moving trajectory of a video object. It is an important video feature, for it shows the spatio-temporal relationship of a video object in a video. A motion track is composed of object positions in a sequence of frames. Each position can be represented as the location of the object's center-of-gravity. In [1], the experiments show how motion tracks can be used to improve the performance of the content-based video retrieval. To manage the motion tracks to provide effective searching and query processing, several approaches have been proposed. Yoshitaka and Hirakawa [7] used chain code of directions to represent motion tracks. Wai and Chen [6] proposed an *index region* approach to support approximate queries and a *peak model* to represent the orientations and angles on the motion track. A finite automata based method is used for efficient query processing. Dagtas [3] proposed a *trail-based model* in which the motion tracks of the video object are represented by trails. In [5], the third-order polynomial is used to represent the motion track, and the sum of the differences of the corresponding coefficients of the polynomials is used as the similarity of the motion tracks.

The MPEG-7 standard [8] aims to provide a standard way for describing the content of multimedia data. Since only the polynomials of X and Y with variable time are defined in MPEG-7, the motion trajectory in the X-Y plane needs to be further derived. Moreover, the similarity of two motion trajectories defined in MPEG-7 is the sum of the distances of the corresponding locations on the two motion trajectories. It does not take the advantage of the polynomial representation. In this paper, a new motion track representation based on the motion trajectory in the X-Y plane and the changes of velocities along the X-axis and Y-axis is proposed. A corresponding similarity measure is defined. Furthermore, we propose a matching method for the motion tracks and a relevance feedback mechanism is also used to improve the efficiency and effectiveness of the query processing. Detail of the approach can be further referenced in [2].

The rest of this paper is organized as follows. Section 2 describes the representation of the motion track. The segmentation of motion tracks, similarity measure, and relevance feedback mechanism are described in Section 3. Some experiment results are shown in Section 4. Section 5 concludes this paper and points out the future work.

2 MOTION TRACK REPRESENTATION

The motion track of a moving object can be represented by the motion trajectory and the velocities. Both of them can be modeled in polynomial. There are several advantages to model the motion trajectory in polynomial [4]. First, it is easy to derive the velocity and the acceleration of an object at some position on the motion trajectory. It is also compact compared

* Corresponding author Email: alpchen@cs.nthu.edu.tw

with recording all the positions of the object on the motion trajectory. Moreover, the compact radio on using how many data points to construct the polynomial can be determined according to the need. In this paper we propose a different method from MPEG-7 motion descriptor to represent the motion trajectory and velocities.

2.1 Representation of motion trajectory

We use the first-, second-, or third-order polynomial of Y with variable X to represent the motion trajectory on the X-Y plane. In the MPEG-7 motion descriptor, only the first- and second-order polynomials are considered. We also use the third-order polynomial because of its low computation time and high flexibility compared with the first- and second-order polynomials for modeling arbitrary curves of motion tracks. The polynomial is constructed by the linear regression method.

2.2 Representation of velocities

In addition to the motion trajectory, we also record the velocities along the X-axis and Y-axis of a moving object, denoted V_x and V_y. V_x and V_y are the derivatives of the regressive curves of the x values and y values along time, respectively. The detail is shown in Figure 1.

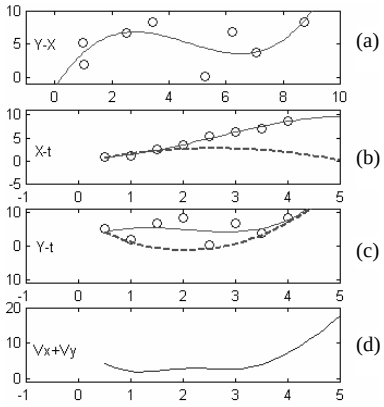

Figure 1 (a) The regressive curve on the X-Y plane, (b) the regressive curve of x along time and its velocity curves V_x (the dotted line), (c) the regressive curve of y along time and the its velocity curve V_y (the dotted line), (d) the velocity combined by V_x and V_y.

3. QUERY PROCESSING

In this section, we introduce our approach of the video retrieval based on the motion tracks of the moving objects. It includes: (i) segmentation of the motion tracks; (ii) similarity measure based on the proposed representation of motion tracks; and (iii) matching the query with the motion tracks in the database.

3.1 Segmentation

For a motion track with a turning behavior, using one single regressive curve may not be able to capture the turning characteristics. This is because the regressive curve only considers the distribution of the object locations. However, it is an important feature of the motion tracks which users may be interested in. For example, users may issue a query: "*Find a video clip in which a bee is flying around.*" How can this turning behavior be described?
Since we use the polynomial of Y along X as the motion trajectory representation, the turning behavior which cannot be described by a regressive curve is the turning behavior along the X-axis. We define a *turning point* along the X-axis as follows. For a sequence of points along the X-axis x_i, i=1, 2,…, m, x_{turn} is a turning point when

(i) $x_1, x_2, \ldots, x_{turn-1} < x_{turn}$ and $x_{turn+1}, x_{turn+2}, \ldots, x_m < x_{turn}$, or
(ii) $x_1, x_2, \ldots, x_{turn-1} > x_{turn}$ and $x_{turn+1}, x_{turn+2}, \ldots, x_m > x_{turn}$.

The motion track with the turning behavior along the X-axis will be segmented on the turning points. Each segment can then be modeled by a regressive curve. Based on this method we can preserve the turning behaviors and precisely represent the motion track.

Figure 2(a) shows a motion track with four turning points. By our segmentation method, the motion track can be segmented into five segments as shown in Figure 2(c).

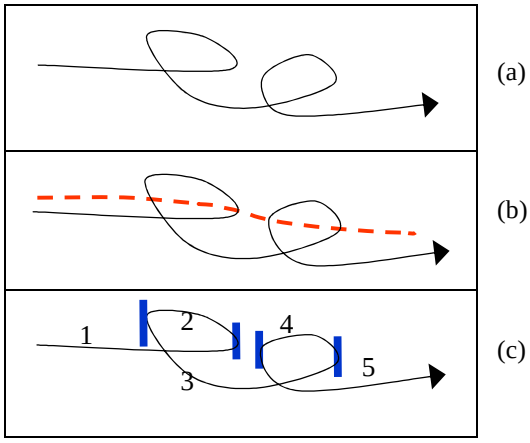

Figure 2 (a) A motion track with a turning behavior, (b) the corresponding regressive curve (the dotted line), and (c) the motion track segmented into five segments on four turning points.

3.2 Similarity Measure

After constructing the motion trajectory and the velocity polynomials, the coefficients and some other meaningful features based on the polynomials will be used to derive the similarity measure. In [5], the sum of the differences of the corresponding coefficients is used to measure the distance between two polynomials of the motion tracks. In this section we define appropriate similarity measures for both the motion trajectory and velocities. Based on the similarity measures of the motion trajectory and velocities, the similarity measure of two motion tracks is defined.

3.2.1 Similarity Features of Motion Trajectories

The motion track is modeled by a first-, second-, or third- order polynomial. For a third-order polynomial $y = ax^3 + bx^2 + cx + d$, the corresponding derivative with x is $y' = 3ax^2 + 2bx + c$. If a curve has two peaks (a local maximum and a local minimum), the polynomial of y' should satisfy the inequality $(2b)^2 - 4 \times 3a \times c \geq 0$. The inequality can be reduced as:

$$b^2 \geq 3ac \quad (1)$$

This inequality is then used to check whether this third-order polynomial is degenerate. In the following, we define the *similarity feature* (SF) for each type of the polynomial and explain how these SFs are derived and the advantages of using these values.

(i) Non-degenerate third-order polynomial

$SF = \left| \frac{y_2 - y_1}{x_2 - x_1} \right|$, where (x_1,y_1) and (x_2,y_2) are the two peaks of the polynomial.

(ii) Degenerate third-order polynomial

$SF = w_1 \times S_{slope} + w_2 \times S_{area}$, where S_{slope} is the slope of the tangent line on the inflection point, and S_{area} is the area between the tangent line on the inflection point R (R_x, R_y) and the cubic curve in the range of $R_x \pm \varepsilon$; also, w_1 and w_2 are weights used for a better combination of the two values.

The degenerate third-order curve does not have a local maximum or local minimum. By observation, the inflection point (the point on which the second derivative is 0) and the *smoothness* of the curve are important features of the degenerate third-order polynomial. These two features are combined with suitable weights to compute the SF. For the variable ε, there exists a tradeoff between the precision and efficiency. Figure 3 illustrates an example to show the slope of the tangent line S_{slope} and the area S_{area}.

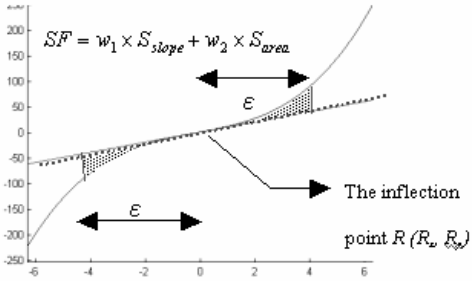

Figure 3 The degenerate third-order polynomial with the tangent line passing through the inflection point.

(iii) Second-order polynomial

SF = the area between the tangent line of the minimum or maximum point $R(R_x, R_y)$ and the curve in the range of $R_x \pm \varepsilon$. In order to know the smoothness of the curve, we use the area between the tangent line on the local maximum or minimum and the curve in some range as the SF. This idea is similar to the SF of the degenerate third-order polynomial.

(iv) First-order line

SF = the slope of the line. For a line, it is simple to use the slope as the SF.

3.2.2. Velocity similarity

In Section 2.1, the velocity of a motion track is represented by two polynomials, V_x and V_y, which are derivatives of X and Y polynomials along time. We consider the velocity similarity in two aspects, i.e., the changes of the velocity and the average velocity. In this paper, we only consider the velocity V_x and V_y as second-order polynomials. For a second-order polynomial, there are six possible velocity trends. We define the similarity between the velocity trends, called *VelSim*, from 0 to 1 to show the similarity of trends. Table 1 and Table 2 list the velocity trends and the corresponding similarity matrix in our video retrieval system.

Decreasing	Increasing	Both
1	2	3
4	5	6

Table 1 Six possible velocity trends.

Type	1	2	3	4	5	6
1	1	0	0.5	0.8	0	0.4
2	0	1	0.5	0	0.8	0.4
3	0.5	0.5	1	0.4	0.4	0

4	0.8	0	0.4	1	0	0.5
5	0	0.8	0.4	0	1	0.5
6	0.4	0.4	0	0.5	0.5	1

Table 2 The similarity matrix of *VelSim*.

3.2.3. Similarity of motion tracks

After defining the SFs of the motion trajectory and the velocity similarity, we combine these values for the similarity between a query segment Q_{SEG} and a motion track segment C_{SEG} in the database. For convenience, we use distance as the similarity measure where a larger distance implies a lower degree of similarity.

$$Dis\,(Q_{SEG}, C_{SEG}) = w_1 \times \left| SF_Q - SF_c \right| + w_2 \times \left[(1 - VelSim_x) + (1 - VelSim_y) \right] + w_3 \times \left| \overline{V}_Q - \overline{V}_C \right| \qquad (2)$$

where SF_Q and SF_C are the similarity features of Q_{SEG} and C_{SEG}; $VelSim_x$ and $VelSim_y$ represent the similarity between the V_x and V_y of the Q_{SEG} and C_{SEG} respectively; $\overline{V}_Q$ and $\overline{V}_C$ are the average velocity of Q_{SEG} and C_{SEG}; and w_1, w_2 and w_3 are the weights for the three components. If Q_{SEG} and C_{SEG} belong to different types, then the $Dis(Q_{SEG}, C_{SEG})$ is set to ∞.

3.2.4 Matching method

For a query, we design an efficient way to match the query track and the motion tracks in the database. We use the k-Nearest-Neighbor (KNN) method to find the k best answers. A two-level KNN method is proposed for the matching of more than one segment. Based on this method, partial query processing is performed and the approximate answers obtained.

The two-level KNN search is described as follows:

Step 1. Segment the query on the turning point. For each query segment, find k nearest-neighbor (NN) segments in the motion track database, and compute the similarity between a query segment and its NN segments by formula (2).

Step 2. Link the NN segments if they belong to the same motion track and are consecutive segments as the results. This is shown in Figure 4.

Step 3. From the results, find the k' longest lists. The length of a list is the number of segments. The number of k' depends on how many answers are retrieved. We choose the k' longest lists first, since each segment in the list is already the best answer of the query segment. Therefore the longer the list is, the better the result will be. For the answers with the same length, we rank them according to their combined dissimilarity defined as follows.

$$\text{Combined dissimilarity of multi-segments} = \frac{\sum Dis(Q_{SEG}, C_{SEG})}{\#\ of\ \ segments},$$

where $Dis(Q_{SEG}, C_{SEG})$ means the dissimilarity of the corresponding segments, and the combined dissimilarity is the average dissimilarity of each segment.

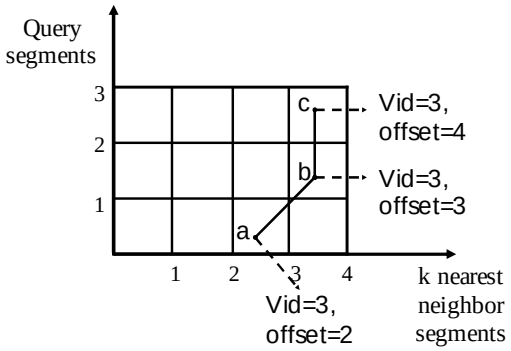

Figure 4 The processing of two-level KNN search.

In Figure 4, the number of rows is the number of query, and the number of columns is the number of NN segments retrieved. In this example, the query has three segments, and for each segment of the query, we retrieve 4 best answers. By Step 2, we check if any two NN segments have the same Vid and are consecutive. In Figure 4, the segment a and b are consecutive because they are the second and third segments of the motion track in the video with Vid=3. So are the segments b and c. We find a result with length 3 as shown in Figure 4.

The disadvantage of this two-level KNN method is that the value of k should be larger, or we cannot find enough answers. However, since the size of the database is usually large, we can prune a lot of candidates in the first-level KNN. Furthermore, some approximations like skipping or inserting one segment can be applied on this method.

3.3 Weight Adjustment by Relevance Feedback

Since whether the motion tracks are similar or not is very subjective, we design a relevance feedback mechanism for users to adjust their answers.

In equation (2), we combined three features to measure the distance between two motion track segments. We adopt the main idea of the standard deviation method in our approach. First, the *positive* and *negative* results are chosen by users. Then the variances of the three features of positive and negative examples are analyzed respectively. For the positive results, the bigger the variance of some feature is, the smaller the feature weight should be. For the negative results, the smaller the variance of some feature is, the smaller the feature weight should be.

Users may be more interested in the velocity trends than the subtle difference of the motion trajectories. Also, it is difficult to handle the variances of the similarity feature. In this paper we focus on the w_2 and w_3 adjustment to help users to find results with similar velocity trends or speeds. The way to adjust the weights w_2 and w_3 in our video retrieval approach is illustrated as follows:

a. For the positive results:

$$w_2 = \begin{cases} \dfrac{w_2}{Pos\_\operatorname{var} V_x \times Pos\_\operatorname{var} V_y \times R^2} \\ \quad \text{if } Pos\_\operatorname{var} V_x \neq 0 \text{ and } Pos\_\operatorname{var} V_y \neq 0 \\ \dfrac{w_2}{Pos\_\operatorname{var} V_x \times R} \times C \text{ if } Pos\_\operatorname{var} V_x = 0 \\ \dfrac{w_2}{Pos\_\operatorname{var} V_y \times R} \times C \text{ if } Pos\_\operatorname{var} V_y = 0 \end{cases} \quad (3)$$

$$w_3 = \frac{w_3}{Pos\_\operatorname{var} Vel} \quad (4)$$

For all positive results, the variances of the V_x and V_y distances are calculated as $Pos\_\text{var}V_x$ and $Pos\_\text{var}V_y$, and the V_x and V_y distances are the *VelSim* distances between the query and the V_x and V_y, respectively. Since the range of the distance is from 0 to 1 and the variance is always less than 1, we re-range it to a constant R to prevent w_2 from always growing larger. If the $Pos\_\text{var}V_x$ or $Pos\_\text{var}V_y$ is equal to 0, it means it is a very important feature for the user. To prevent from dividing-by-0 and enhance the effect of this feature, we multiply w_2 by a constant C.

In formula (4), the w_3 is re-weighted by dividing w_3 by the variance of the average velocity of the positive results.

b. For the negative results

The idea of the negative weight adjustment is similar to the positive one. However the effect of the variance is different. The smaller the variance of the feature is, the smaller the weight should be.

Table 3 shows the three experiments and corresponding results. The experiment (a) is to find top 8 best results in 30 second-order curves, and select one of *Vx*, *Vy* and average velocity as the feature that users are interested in. For example, we want to find the similar curves with their velocities along X-axis being speeding up, or the average velocity ranging from 1.5 to 2 km/sec. The precision is measured by the percentage of positive results in the top 8 results. The first column is the precision and the second column is the rank of the best match curve. After the user sends the feedback, the weights will be adjusted according to formulas (3) to (6). The query will be processed again with new weights, and then the precision and the rank of the best match are measured. From Table 4 we can see the precisions improve a lot in the second query, but just a little in third query. This means our weight adjustment approach is efficient and in most cases users can get the results they want. In the experiments (a) and (b) only the positive effects are considered, and in (c) both the positive and negative effects are considered. Comparing (b) with (c) we can find using both positive and negative weight adjustments is more precise than only using the positive weight adjustment.

	Initial Query precision	Rank of best match of initial query	Second Query precision	Rank of best match of 2nd query	Third Query precision	Rank of best match of 3rd query
(a) 2nd curve with pos.	0.55	1.4	0.825	1.1	0.825	1.1
(b) 3rd curve with pos.	0.6125	1.5	0.7625	1.3	0.7875	1.3
(c) 3rd curve with pos. & neg.	0.6	1.2	0.825	1.2	0.825	1.2
Average of the 3 experiments	0.5875	1.367	0.804	1.2	0.8125	1.2
Improvement rate	---	---	0.369	0.122	0.0104	0

Table 3 Results of query refinement experiments. Select 8-NN results and C=10, R=3.

4. EXPERIMENT RESULTS

In order to evaluate our approach, we built a video retrieval system based on the motion tracks. We analyze the performance of our video retrieval system in both the efficiency and effectiveness.

We compare our approach with [5] which also uses the third-order polynomial to represent a motion track. The differences are listed in Table 4.

	Lee's work [5]	Our approach
(i) segmentation	No	Segment on the turning point
(ii)trajectory representation	3rd-order polynomial	1st/ 2nd/ 3rd polynomial
(ii)velocity representation	average velocity	V_x, V_y and average velocity
(iii)trajectory distance measure	The sum of differences of corresponding coefficients	Divide motion tracks into several types, and define SF to measure distance

(iv) velocity distance measure	The difference of average velocities	The difference of average velocities and the trends of V_x and V_y
(vi) indexing	No	Divide into several types and cluster according to SFs

Table 4 The difference between [5] and our approach.

Table 5 shows the effectiveness comparison of [5] and our approach. The precisions and the ranks of the most similar results are compared. The precision is measured by the number of relevant results out of the number of the retrieved results. Moreover, the most similar result is chosen by users. In our approach, the precision is about 85%, and most of the irrelevant results come from unsuitable regressive curves. Therefore, if we want to improve the precision, stricter criteria to construct the suitable regressive curve are needed. In [5], the precision is about 50% and the average rank of most similar results is 1.7. It means that similar motion tracks may have similar coefficients, but motion tracks with similar coefficients may not be similar.

Precision of our approach	most similar rank	Precision of [5]	most similar rank
0.844	1.067	0.505	1.733

Table 5 The precisions of our approach vs. [5].

5. CONCLUSIONS

In this paper, we propose a new descriptor to represent the motion track in the X-Y plane and the trend of velocities. The polynomials are used to model the motion trajectory and velocities. The similarity measure based on the properties of the polynomial is defined for comparing two motion tracks. A novel motion track segmentation method is also proposed to handle a complex motion track with turning behavior. Moreover, the relevance feedback is used to improve the query results. By the experiment results we show that our approach is more effective than existing ones using polynomial representations. We will continue to improve the similarity feature such that the SF values from all types of the polynomials can be normalized. Moreover, it is important to find a better way to construct regressive curves more appropriately to improve the precision of the query results.

REFERENCES

[1] S.F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Transactions on Circuit and Systems for Video Technology*, Vol. 8, No. 5, pp. 602-615, 1998.

[2] P.Y. Chen and Arbee L.P. Chen, "Video Retrieval Based on Similarity of Motion Tracks of Moving Objects," TR-MAKE-002, National Tsing Hua University, Taiwan, 2002.

[3] S. Dagtas, W. Al-Khatib, A. Ghafoor, and R. L. Kashyap, "Models for Motion-Based Video Indexing and Retrieval," *IEEE Transactions on Image Processing*, Vol. 9, No. 1, pp. 88-101, 2000.

[4] S. Jeannin and A. Divakaran, "MPEG-7 visual motion descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 6, pp.720-724, 2001.

[5] K. Lee, W. You and J. Kim, "Video Retrieval based on the Object's Motion Trajectory," *Visual Communications and Image Processing*, 2000.

[6] T. Y. Wai and Arbee L. P. Chen, "Retrieving Video Data via Motion Tracks of Content Symbols," *Proc. ACM International Conference on Information and Knowledge Management*, 1997.

[7] A. Yoshitaka, M. Yoshimitsu, M. Hirakawa, and T. Ichikawa, "V-QBE: Video Database Retrieval by Means of Example Motion of Objects," *Proc. IEEE International Conference on Multimedia Computing and Systems*, 1996.

[8] ISO/IEC JTC1/SC29/WG11 N4509, "Overview of the MPEG-7 Standard", Pattaya, December 2001.

[9] M.Garcia and H.Nicolas, "Video Object Trajectory Analysis," ICIP 2003.