Subtrajectory-Based Video Indexing and Retrieval

Thi-Lan Le[1,2], Alain Boucher[1,3], and Monique Thonnat[2]

[1] International Research Center MICA
Hanoi University of Technology, Viet Nam
Thi-Lan.LE@mica.edu.vn, alain.boucher@auf.org
[2] ORION, INRIA, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis, France
{Lan.Le_Thi, Monique.Thonnat}@sophia.inria.fr
[3] Equipe MSI, Institut de la Francophonie pour l'Informatique, Hanoi, Viet Nam

Abstract. This paper proposes an approach for retrieving videos based on object trajectories and subtrajectories. First, trajectories are segmented into subtrajectories according to the characteristics of the movement. Efficient trajectory segmentation relies on a symbolic representation and uses selected control points along the trajectory. The selected control points with high curvature capture the trajectory various geometrical and syntactic features. This symbolic representation, beyond the initial numeric representation, does not suffer from scaling, translation or rotation. Then, in order to compare trajectories based on their subtrajectories, several matching strategies are possible, according to the retrieval goal from the user. Moreover, trajectories can be represented at the numeric, symbolic or the semantic level, with the possibility to go easily from one representation to another. This approach for indexing and retrieval has been tested with a database containing 2500 trajectories, with promising results.

1 Introduction

Advances in computer technologies and the advent of the World Wide Web have made an explosion of multimedia data being generated, stored, and transmitted. For managing this amount of information, one needs developing efficient content-based retrieval approaches that enable users to search information directly via its content. Currently, the most common approach is to exploit low-level features (such as colors, textures, shapes and so on). When working with videos, motion is also an important feature. When browsing a video, people are more interested in the actions of a car or an actor than in the background. Moving objects attract most of users' attention. Among the extracted features from object movement, trajectory is more and more used. In order to use the trajectory information in content-based video indexing and retrieval, one must have an efficient representation method allowing not only to index trajectories, but also to respond to the various kinds of queries and retrieval needs. For retrieval aspects, the matching strategies is also of importance.

Matching to compare between trajectories can be done globally or partially. For global matching, the whole trajectories are compared to each other. However, objects in videos can undergo complex movements, and global matching can prevent from retrieving a partial but important section of the trajectory. In some cases, the user can be interested in only one part of the object trajectory. Therefore, it is useful to segment a trajectory into several subtrajectories and then match the subtrajectories. How to match subtrajectories and how to combine all partial results into a final retrieval result is not as trivial as it seems. In [1], the authors have segmented the object trajectory into subtrajectories with constant acceleration. But this approach do not consider the case where the object changes direction. In [2], after segmenting a trajectory into subtrajectories, the authors computed the PCA coefficients for each subtrajectory. However, with only one matching strategy, it cannot satisfy all user needs.

In the retrieval phase, queries from the user can be done at the numeric, symbolic or semantic level. Many interaction levels allow the user to be more flexible regarding his/her needs. Similarly, distances between a query and the database can be measured according to one level or another. Another important aspect in interactive retrieval is the relevance feedback, which allows an user to refine the query. A good representation scheme and efficient matching strategies must take into account all these aspects, taking care of both indexing and retrieval.

After introducing all these aspects, the main contributions of this paper are the following: Integrate into a representation scheme numeric, symbolic and semantic levels for trajectory-based video indexing and retrieval; Propose a trajectory segmentation algorithm working at the symbolic level, to avoid the problem of sensibility to noise at the numeric level, and invariant to rotation, translation or scaling; Present different trajectory and subtrajectory matching strategies; Go toward semantic trajectory-based video indexing and retrieval.

The rest of the paper is organized as follow. In Section 2, we are proposing a structure for a SubTrajectory-based Video Indexing and Retrieval (STBVIR), which includes control point selection, trajectory representation at the numeric and the symbolic level, trajectory segmentation into subtrajectories and matching strategies. In section 3 we are presenting some aspects linked to semantic. Some experimental results are shown in section 4. Section 5 is concluding this paper with some directions for future work.

2 SubTrajectory-Based Video Indexing and Retrieval

2.1 General description

We are proposing an architecture of STBVIR (figure 1). In this architecture, object tracking is done by a preprocessing module (not shown here), and object trajectories are taken as input. In the real physical world, a trajectory is represented following 3 dimensions. Without a priori contextual information, trajectories can be represented in 2D. Knowing the application and its context, it can be useful to represent a trajectory in 3D or to map the 2D trajectory into the monitored environment [3]. In this paper, we will consider only the general case in 2D, without a priori knowledge on the application.

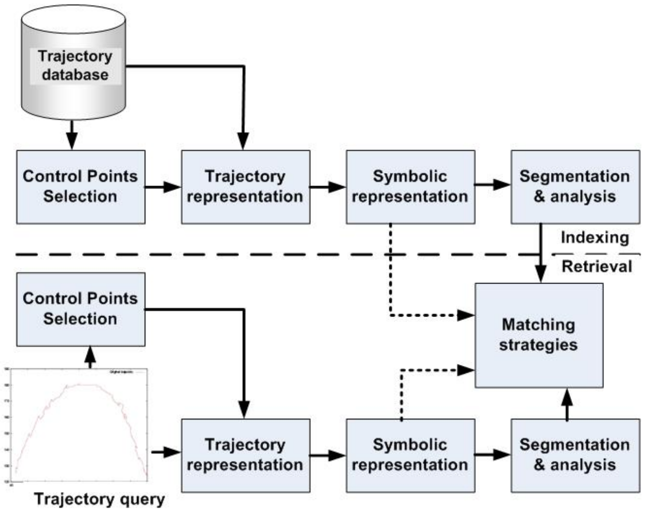


Fig. 1. Architecture for subtrajectory-based video indexing and retrieval

For indexing, all object trajectories are processed through four modules. The output is a symbolic representation of the global trajectory or its subtrajectories or only some selected control points along the trajectory. For retrieval, given a trajectory query by the user, comparison is made with the trajectories in the database, at the numeric or the symbolic level. Trajectories that are most similar (given a matching strategy) with the trajectory query will be returned to users.

2.2 Control point selection

A symbolic representation can take all the individual trajectory points as input. But doing so, computation time can be high, as well for the symbolic representation as for the trajectory matching. Selecting control points with high curvatures along the trajectory before computing its representing can help greatly. Selected control points can capture the trajectory's various geometrical and syntactic features. As one can see in section 4, the results from the two cases, using all points or only some selected control points, are very similar, but the second case takes much less time to compute. Moreover, selected control points and their symbolic representation allow us to propose a segmentation method as described in the next section.

First, we are describing the control point selection method of [4]. Given a sequence, $T = [(x_1, y_1), ..., (x_n, y_n)]$, n being the length of T, T can be represented by $T = [p_1, .., p_n]$. Let $\alpha(p)$ be the angle of a point p in T, determined by

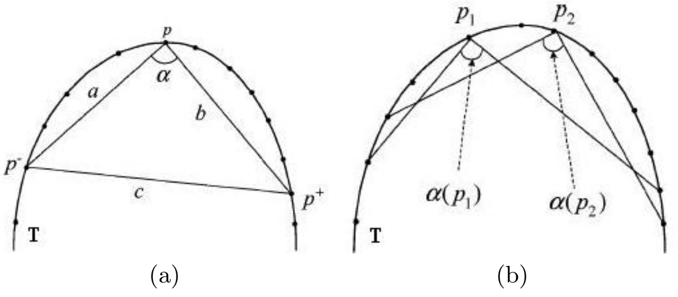


Fig. 2. Selecting control points along a trajectory T. (a) Control points (like p) are shown along the trajectory T. p^- and p^+ are linked to the point p by satisfying the angle constraint given by the Equation 1. (b) p_1 and p_2 are two selected control points too close to each other. The one with the smaller angle α will be chosen as the best control point [4].

two specified points p^+ and p^- which are selected from both sides of p along T (figure 2.a) and satisfy

$$d_{min} \leq |p - p^+| \leq d_{max} \text{ and } d_{min} \leq |p - p^-| \leq d_{max} \tag{1}$$

where d_{min} and d_{max} are two thresholds. d_{min} is a smoothing factor used to reduce the effect of noise from T. With p^+ and p^-, the angle can be computed using

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

If $\alpha(p)$ is larger than a threshold T_α, set to 150 here, the point p is selected as a control point. In addition to equation 2, it is expected that the two control points are far from each other, to enforce that the distance between any two control points is larger than the threshold defined in (1). If the two candidates are too close to each other, i.e. $||p_1 - p_2|| \leq d_{min}$, the one with the smaller angle α is chosen as the best control point (figure 2.b).

2.3 Numeric trajectory representation module

Working with raw data from the object trajectory is not always suitable because these data are sensible to noise and are affected by rotation, translation and scaling. In order to cope this problem, we have chosen among the existing representation methods the one from [5], which also uses both direction and distance information of the movement.

With a given sequence, $T_A = [(x_{a,1}, y_{a,1}), ..., (x_{a,n}, y_{a,n})]$, n being the length of T_A, a sequence of (movement direction, movement distance ratio) pairs M_A is defined as a sequence of pairs: $M_A = [(\theta_{a,1}, \delta_{a,1}), ..., (\theta_{a,n-1}, \delta_{a,n-1})]$. The movement direction is defined as:

$$\theta_{a,i} = \begin{cases} \arctan \frac{y_{a,(i+1)} - y_{a,(i)}}{x_{a,(i+1)} - x_{a,(i)}} - \pi & \text{if } x_{a,(i+1)} - x_{a,(i)} \prec 0 \text{ and } y_{a,(i+1)} - y_{a,(i)} \leq 0 \\ \arctan \frac{y_{a,(i+1)} - y_{a,(i)}}{x_{a,(i+1)} - x_{a,(i)}} & \text{if } x_{a,(i+1)} - x_{a,(i)} \geq 0 \\ \arctan \frac{y_{a,(i+1)} - y_{a,(i)}}{x_{a,(i+1)} - x_{a,(i)}} + \pi & \text{if } x_{a,(i+1)} - x_{a,(i)} \prec 0 \text{ and } y_{a,(i+1)} - y_{a,(i)} \succ 0 \end{cases} \tag{3}$$

and the movement distance ratio is defined as follows:

$$\delta_{a,i} = \begin{cases} \frac{\sqrt{(y_{a,(i+1)} - y_{a,(i)})^2 + (x_{a,(i+1)} - x_{a,(i)})^2}}{TD(T_A)} & \text{if } TD(T_A) \neq 0 \\ 0 & \text{if } TD(T_A) = 0 \end{cases} \tag{4}$$

$$TD(T_A) = \sum_{1 \leq j \leq n-1} \sqrt{(y_{a,(j+1)} - y_{a,j})^2 + (x_{a,(j+1)} - x_{a,j})^2} \tag{5}$$

Raw trajectory data given to this module become a sequence of pairs of movement direction and movement distance ratio. We can use directly this sequence to compare trajectories or we can use it as an intermediate information for the symbolic representation module.

2.4 Symbolic trajectory representation module

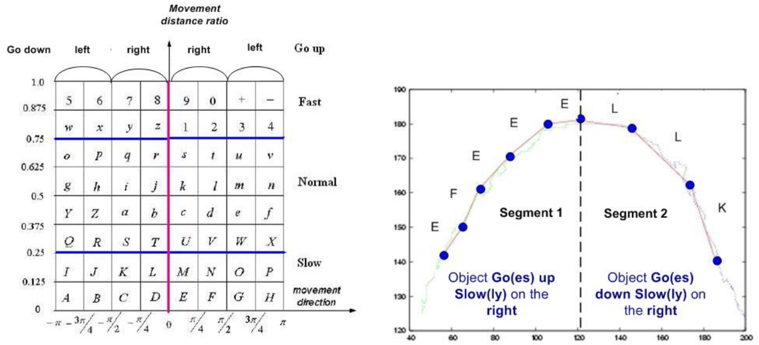
Using the previous numeric representation for trajectories, a proposed symbolic representation from [5] is computed as follows:

Given ϵ_{dir} and ϵ_{dis}, the two dimensional (movement direction, distance ratio) space is divided into $\frac{2\pi}{\epsilon_{dir}} * \frac{1}{\epsilon_{dis}}$ subregions. Each subregion SB_i is represented by two (movement direction, distance ratio) pairs: $(\theta_{bl,i}, \delta_{bl,i})$ and $(\theta_{ur,i}, \delta_{ur,i})$, which are the bottom left and upper right coordinates of SB_i. A distinct symbol A_i is assigned for subregion SB_i of size $\epsilon_{dir} * \epsilon_{dis}$. A pair of movement direction and movement distance ratio $(\theta_{a,i}, \delta_{a,i})$ will be represented by a symbol A_i if $\theta_{bl,i} \leq \theta_{a,i} \prec \theta_{ur,i}$ and $\delta_{bl,i} \leq \delta_{a,i} \prec \delta_{ur,i}$.

2.5 Segmentation

From an original trajectory composed of n points $T = [(x_1, y_1), ..., (x_n, y_n)]$, a new trajectory, shorter than the original one, is obtain after control point selection: $T' = [(x_1, y_1), ..., (x_m, y_m)]$ where $m \leq n$. This trajectory is transformed into a symbolic representation $S = [(A_1, ..., A_m)]$ using the quantization map shown in figure 3.a.

Let A_I be the set of symbols with θ being smaller than 0 and A_{II} be the set of symbols with θ greater than 0. An object going down gets a symbol belonging to A_I (down to the left or to the right), and an object going up gets a symbol belonging to A_{II} (up to the left or to the right). Therefore, by scanning a symbolic representation until a change in direction is detected (i.e. symbol at time t belongs to A_I and symbol at time t+1 belongs to A_{II}, or the opposite), we can create a new subtrajectory including all the points from the last change in direction to this new change, and so on until the end of the trajectory.



(a) (b)

Fig. 3. (a) Quantization map used for symbolic representation with their corresponding semantic expressions. (b) An example of parsing from a symbolic to a semantic representation using this map.

2.6 Subtrajectory-based video indexing and retrieval

Let $T_Q = [(x_1, y_1), ..., (x_n, y_n)]$ be a query trajectory. Following the segmentation algorithm that we have described in section 2.5, we can segment it in N subtrajectories $T_Q = \{T_{1Q}, ..., T_{NQ}\}$. Let T_D be a trajectory from the indexed databases divided into M subtrajectories $T_D = \{T_{1D}, .., T_{MD}\}$. With all these subtrajectories, we compute their numeric and their symbolic representations. Doing so, we can compare them using either the Edit Distance on Real Sequence (EDR) on the numeric representation or the Edit Distance on Movement Pattern String (EDM) on the symbolic representation [5].

For subtrajectory-based video indexing and retrieval, choosing a good and efficient matching strategy is important to take into account the various user needs. If the user is interested in the whole trajectory, a global matching strategy is a valuable choice, and if he/she just makes more attention in few parts of the trajectory, partial matching is then the privileged choice. Inspiring ourselves from [1], we are giving here some different matching strategies: two global matching strategies (dominant segment (GD) and full trajectory (GF) matching) and three partial trajectory matching strategies (strict partial (SP), relative partial (RP) and loose partial (LP) matching). In the following, $d_{Dist}(T_{iQ}, T_{jD})$ is the distance between a subtrajectory T_{iQ} and a subtrajectory T_{jD}. $Dist$ can be EDR or EDM. Note that L_{iQ} and L_{jD} are the length of T_{iQ} and T_{jD} respectively.

- **Global trajectory matching**
  - ***Dominant segment matching***(GD) Only the dominant subtrajectory is used to match with those in the database. Dominant subtrajectories can be identified as segments with the smallest EDR or EDM distances.

$$d_{Dist}(T_Q, T_D) = min(d_{Dist}(T_{iQ}, T_{jD})) \qquad (6)$$

- ***Full trajectory matching***(GF): All subtrajectories in the original trajectory must match those in the database as described above.

$$d_{Dist}(T_Q, T_D) = \sum_{i=1}^{N} \sum_{j=1}^{M} w_{T_{iQ}} w_{T_{jD}} d_{Dist}(T_{iQ}, T_{jD}) \qquad (7)$$

$$\text{where } w_{T_{iQ}} = \frac{L_{iQ}}{\sum_{k=1}^{N} L_{kQ}} \text{ and } w_{T_{jD}} = \frac{L_{jD}}{\sum_{k=1}^{M} L_{kD}} \qquad (8)$$

– **Partial trajectory matching**: A subset of subtrajectories is selected to match those in the database, and matching is specified in terms of order. Matching between two subtrajectories is computed using:

$$match_{Dist}(T_{iQ}, T_{jD}) = true \text{ if } d_{Dist}(T_{iQ}, T_{jD}) \leq Threshold_{Dist} \qquad (9)$$

$$match_{Dist}(T_{iQ}, T_{jD}) = false \text{ otherwise} \qquad (10)$$

- ***Strict partial matching***(SP): The matched subtrajectories between the query and the database must be strictly in the same order.
- ***Relative partial matching***(RP): The relative order of the matched subtrajectories between the query and the database must be the same.
- ***Loose partial matching***(LP): No constraint is given on the order of the matched subtrajectories. Just match a subset of subtrajectories between the query and the database.

3 Toward a semantic trajectory-based video indexing and retrieval

The word semantic is more and more used in the information retrieval domain (in many different ways). The given symbolic representation and segmentation method allow us to go toward a more semantic subtrajectory-based video indexing and retrieval. Using the quantization map of figure 3.a, a sequence of symbols representing a trajectory can be translated into a sequence of semantic words, as shown in figure 3.b.

In order to transform a symbolic representation of a trajectory into a semantic representation, some abstraction heuristics must be used. For example, if more than 80% of the symbols of a trajectory belong to the set {'M','N','E','F'} (figure 3.a) it can be said that this trajectory has the 3 characteristics {***Go up***, ***Slow***, ***right***}. In the example of 3.b, the original trajectory is segmented into two subtrajectories. After the symbolic representation phase for the selected control points, the trajectory is represented by 8 symbols 'EFEEELLK'. The first subtrajectory has 5 symbols (EFEEE) while the second one has 3 symbols (LLK). This trajectory can be abstracted saying that first the objet ***Go(es) up Slow***ly on the ***right*** and then it ***Go(es) down Slow***ly still on the ***right***.

Using this scheme, the user can give a query at the numeric, symbolic or semantic level. Comparison between (sub)trajectories can be done so far at the numeric (EDR distance) or at the symbolic (EDM distance) level (figure 4.a).

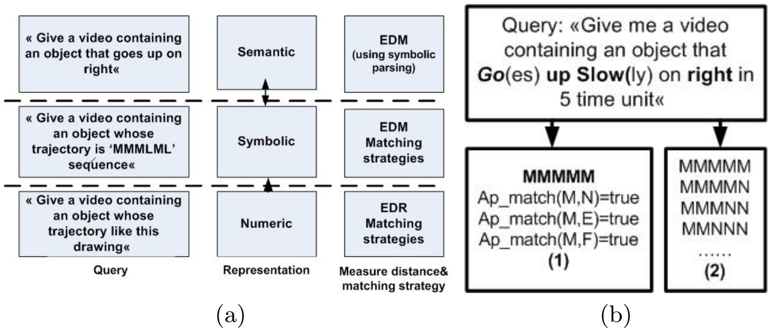

« Give a video containing an object that goes up on right«
Semantic
EDM (using symbolic parsing)
« Give a video containing an object whose trajectory is 'MMMLML' sequence«
Symbolic
EDM Matching strategies
« Give a video containing an object whose trajectory like this drawing«
Numeric
EDR Matching strategies
Query
Representation
Measure distance& matching strategy
(a)
Query: «Give me a video containing an object that Go(es) up Slow(ly) on right in 5 time unit«
MMMMM
Ap_match(M,N)=true
Ap_match(M,E)=true
Ap_match(M,F)=true
(1)
MMMMM
MMMMN
MMMNN
MMNNN
……
(2)
(b)

Fig. 4. (a)Three levels for trajectory representation. At the numeric level, the user draws a query, the EDR distance and corresponding matching strategies are used. At the symbolic level, the user gives a query using a sequence of symbols, the EDM distance and the proposed matching strategies are used. At the semantic level, the semantic query is first parsed into a symbolic query. Then, the EDM distance and the symbolic matching strategies are used. (b) A given semantic query is parsed into a symbolic representation following two different parsing methods.

To process a semantic query, one must first parse it into a symbolic representation. But more than one symbol can correspond to a sole semantic word. For this reason, two methods are possible for parsing the semantic query. The first one is to choose a representative symbol from a semantic characteristic of the movement, using an *Ap_match* comparison like in [5]. A_i *Ap_match* A_j if and only if $A_i = A_j$ or A_i is neighboor of A_j. The second method is to generate all possible symbolic sequences and then, combine or choose between all the matching results (following some pre-defined strategies). In figure 4.b, given the semantic query "Give me a video containing an object that ***Go(es) up Slow***ly on the ***right*** in 5 time units", then the two corresponding symbolic sequences, according to both parsing methods, are shown.

This preliminary discussion about semantic aspect in subtrajectory-based video indexing and retrieval is still on-going work, but we can foreseen some promising results in achieving semantic indexing and retrieval.

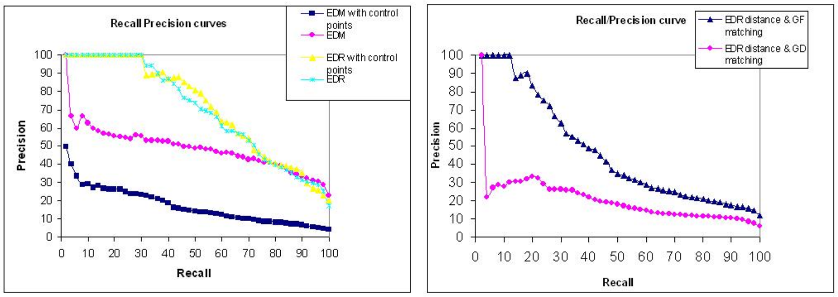
4 Experiments and results

In order to analyze our system performances, we have used the free trajectory database [4] containing 2500 trajectories coming from 50 categories.

Recall and precision curves are widely used to evaluate performance of information retrieval system. In our tests, we have set $n/20$ and $n/15$ for d_{min} and d_{max} respectively where n being the length of T. We have chosen $\epsilon_{dir} = \pi/4$ and $\epsilon_{dis} = 0.125$ for the symbolic representation. Figure 5.a shows the results of our system with all individual of trajectory or some selected control points using the

[4] http://mmplab.eed.yzu.edu.tw/trajectory/trajectory.rar

EDR distance for numeric representation and the EDM distance for symbolic representation. One can realizes that results using selected control points with the EDR distance are comparable with those using all points from the trajectory. The EDR distance in both cases gives better results than the EDM distance.



(a) (b)

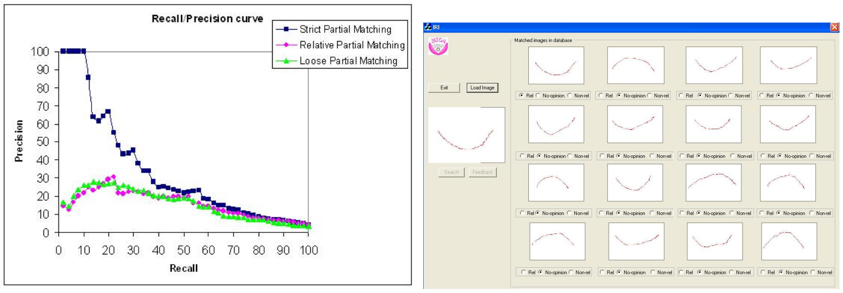
Fig. 5. (a) Recall/precision curve for the EDR and EDM distance with all points from the trajectory and only the selected control points. The curve with green stars and the one with yellow triangles present retrieval results using the EDR distance with all points from trajectories and only selected control points respectively. The curve with violet circles and the one with blue rectangles present retrieval results using the EDM distance with all points from trajectories and only selected control points respectively (b) Recall/precision curve for the EDR distance with different matching strategies. The curve with triangles presents retrieval results using the EDR distance with the Full trajectory (GF) matching strategy while the curve with circles presents retrieval results using the EDR distance with the Dominant segment (GD) matching strategy

Figure 5.b shows the results of our system with different global matching strategies, GF and GD with the EDR distance. With this database, the results with the GF matching strategy are better than with the GD matching strategies. However, in the case where the user is interested in only one part of trajectory, the GD matching strategy is an efficient choice. Figure 6.a shows the results for the three partial matching strategies, SP, RP and LP, using a threshold of 60 and the EDR distance.

Query acquisition and result display is an important but difficult task in trajectory-based video indexing and retrieval. We have implemented a retrieval interface, as shown in figure 6.b. The trajectory query drawn on the left and the first sixteen result images on the right are shown sorted with their EDR distance with the trajectory query. It is possible to draw the corresponding trajectories from a numeric or a symbolic representation.

5 Conclusions and future work

In this paper, a subtrajectory-based video indexing and retrieval system has been proposed. Our system has some notable characteristics. Firstly, it allows the users



(a) (b)

Fig. 6. (a) Results for the SP, RP, LP matching strategies. The curve with green triangles presents the retrieval results using the EDR distance with the SP strategy, the curve with violet circles presents retrieval results using the EDR distance with the RP matching strategy and the curve with blue rectangles presents retrieval results using the EDR distance with the LP matching strategy (b) System interface for retrieval, the trajectory query being on the left and the first sixteen result images on the right shown according to their EDR distance with the trajectory query.

to search desirable trajectory or only part of a trajectory (according to many matching strategies). It effects both EMD and EDR distance that count similar subsequences and assign penalties to the gaps in between these subsequences. Thus, unlike Longest Common SubSequence (LCSS)[5], it does consider gaps within sequences. Secondly, it offers a fast searching (because the trajectories or their subtrajectories are compared by matching only their selected control points), and it allows an efficient segmentation method based on a symbolic representation that is invariant to rotation, scaling and translation. Finally, all these advantages allows us to go toward semantic trajectory-based video indexing and retrieval system, although further work is needed to fully achieve it.

References

1. Chenn, W., Chang, S.F.: Motion trajectory matching of video objects. In SPIE 2003.
2. Bashir, F.: Object Motion Trajectory-Based Video Database System Indexing, Retrieval, Classification, and Recognition. PHD Thesis of Electrical and Computer Engineering, College of Engineering, University of Illinois Chicago (2005).
3. Piciarelli, C., Foresti, G.L., Snidara, L.: Trajectory clustering and its applications for video surveillance.Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2005,40– 45.
4. Hsieh, J. W., Yu, S. L., Chen, Y.S.: Motion-Based Video Retrieval by Trajectory Matching. Proc IEEE Trans. on Circuits and Systems for Video Technology, Vol. 16, No. 3, March 2006.
5. Chen, L., Oŝu, M.T., Oria, V.: Symbolic Representation and Retrieval of Moving Object Trajectories. In MIR'04, NewYork (2004), 15–16