Content-based Video Retrieval via Motion Trajectories

Man-Kwan Shan* and Suh-Yin Lee
Institute of Computer Science and Information Engineering
National Chiao Tung University, HsinChu, Taiwan, ROC

ABSTRACT

Motion is one of the most prominent features of video. For content-based video retrieval, motion trajectory is the intuitive specification of motion features. In this paper, approaches for video retrieval via single motion trajectory and multiple motion trajectories are addressed. For the retrieval via single motion trajectory, the trajectory is modeled as a sequence of segments and each segment is represented as the slope. Two quantitative similarity measures and corresponding algorithms based on the sequence similarity are presented. For the retrieval via multiple motion trajectories, the trajectories of the video are modeled as a sequence of symbolic pictures. Four quantitative similarity measures and algorithms, which are also based on the sequence similarity, are proposed. All the proposed algorithms are developed based on the dynamic programming approach.

Keywords: content-based retrieval, video retrieval, motion trajectory, similarity measures, sequence similarity

1. INTRODUCTION

Video access is one of the important design issues in the development of multimedia information system, video-on-demand and digital library. Video can be accessed by attributes of traditional database techniques, by semantic descriptions of traditional information retrieval technique, by visual features and by browsing. Access by attributes of database technique and access by semantic descriptions of information retrieval technique are insufficient for video access owing to the numerous interpretation of visual data. Besides, the automatic extraction of semantic information from general video programs is outside the capability of the current technologies of machine vision.

Access by visual features retrieves video clips on the basis of content features. Research on content-based video retrieval classifies visual features into spatial and temporal features. The approach of spatial features selects a set of representative key frames from each shot. Image retrieval techniques are then applied on the extracted key frames. Similarity between two shots is derived by comparing visual features of the most similar key frames between them. Temporal features include statistical visual features over all frames in a shot, statistical motion features derived from optical flows, camera work and object motion trajectory.

Among all, object motion trajectory is the most intuitive specification of motion features for human beings. The object may be a recognized object, a region or a collection of regions exhibiting consistency across several video frames.[2] Users can access video clips by specifying objects and associated motion trajectories or just sketching the motion trajectory. Two distinguished features of retrieval via motion trajectories lie in the sub-matching and approximate matching.[9] For sub-matching, users query is the subpart of the trajectory of accessed video. For approximate matching, it is impossible for users to specify the query trajectory precisely.

Previous approaches modeled the trajectory matching problem as the pattern matching problem.[4, 5, 8, 9, 10] The response to the query only returns the qualified video sequences, not relevant video sequences along with degrees of relevance.

In this paper, quantitative similarity retrieval via single motion trajectory and multiple trajectories are addressed. For the retrieval via single motion trajectory, the trajectory is modeled as a sequence of segments while each segment is represented as the slope. Two quantitative similarity measures OCM and OCMR are proposed for approximate matching between query and video trajectory.

For the retrieval via multiple motion trajectories, the simple approach is the intersection of results of individual query motion trajectory. However, this simple approach considers neither the spatial nor the temporal relationships between the query motion trajectories. We follow the approach as suggested by Chang et al.[2] Multiple motion trajectories are modeled as a sequence of symbolic pictures while each symbolic picture is represented as a 2D string. Four quantitative similarity measures based on sequence similarity are proposed.

This paper is organized as follows. In the next section, some work related to the motion trajectory matching is reviewed. In section 3, two proposed similarity measures and algorithms for retrieval via single motion trajectory are presented. In section 4, we discuss the retrieval via multiple motion trajectories and define four quantitative video similarity measures and

* To whom all correspondence to be sent. E-mail: shan@cc.nctu.edu.tw.

algorithms. Section 5 describes the normalization of the proposed similarity measures. Conclusions and future work are described in section 6.

2. RELATED WORK

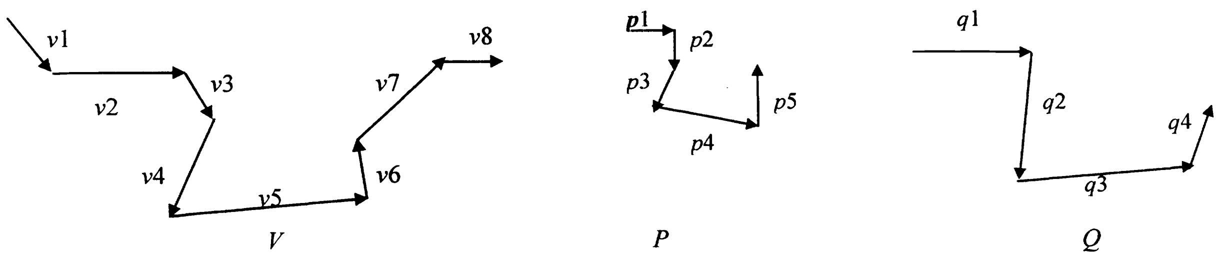

Figure 1. Examples of motion trajectories.

Table 1. Chain-coding scheme for motion trajectory representation.

Primitive	→	↗	↑	↖	←	↙	↓	↘
Code	1	2	3	4	5	6	7	8

Dimitrova and Golshani presented several mechanisms for representation of motion trajectory.[4,5] These mechanisms include B-spline curve representation, chain-coding, differential chain-coding. For example, using the chain-coding scheme illustrated in Table 1, the video motion trajectory *V* in Figure 1 is represented as (81862421). The query motion trajectory *P* is represented as (17683). Matching functions used for motion retrieval depend on the method employed for trajectory representation. For the example of chain-coding scheme, the problem of trajectory matching is translated to the pattern matching problem. However, the matching methods are not discussed.

Notice that exact substring matching algorithms are not suitable for matching of chain-coding scheme. For example in Figure 1, intuitively, given the query trajectory *P*, the video trajectory *V* is somewhat relevant. Using exact substring matching algorithm, *P* represented as (17683) is not the substring of *V* represented as (81862421). Therefore, it is necessary to employ the approximate substring matching algorithms.

Yoshitaka et al. proposed a motion trajectory method based on the index of chain coding scheme.[10] An exhaustive search is applied to map the index motion trajectory to the query curve. However, the exhaustive mapping of the motion trajectory is inefficient.

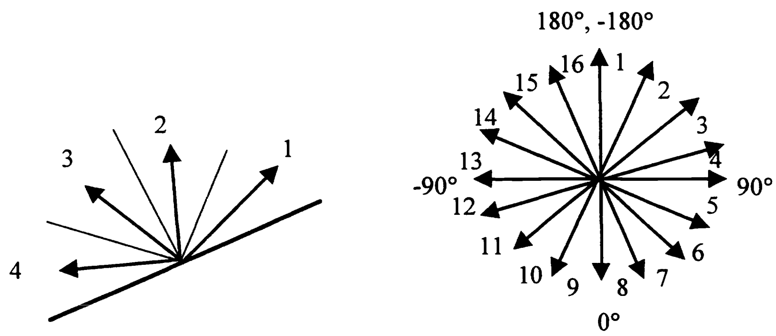

Figure 2. Peak orientation and angle primitives.

Wai and Chen modeled the motion trajectory and the query curve as a combination of peaks.[9] The motion trajectory query processing problem is converted into the string matching problem. Each peak is represented as a triplet, (o_p, a_p, t_t), where o_p, a_p, t_t represent the orientation, angle and time of the peak *p*, respectively. The orientation of the peak is defined as the orientation of the angle bisector and is coded according to the primitives shown in Figure 2. The angle of the peak is defined as the angle expanded by the two edges of the peak and is coded by dividing -180°~ 180° into 16 partitions shown in Figure 2. In order to quantify the similarity for approximate trajectory matching in the peak model, the similarity model is built on the combination code of the angle and the orientation of the peak. Moreover, to deal with the case where, for example, two angles of peak a_p, a_q equal 1 and 16 respectively, the difference, $d(a_p, a_q)$, between a_p, a_q is defined as follows,

If $(a_p < a_q)$ **then** $d(a_p, a_q) = (a_p - a_q) \bmod 16$
else $d(a_p, a_q) = [16 - (a_{ap} - a_a)] \bmod 16$.

Wai and Chen proposed a new finite automata based matching algorithm for string matching which is shown to be superior to UNIX built-in utility *grep*. For example, in Figure 1, the trajectory V is represented as [(1, 15), (2, 14), (4, 14), (1, 7), (3, 5), (4, 15), (3, 15)], while query trajectory P is represented as [(1, 12), (4, 3), (1, 6), (3,6)]. It can be seen that peaks of P (1, 2), (4, 3), (1, 6), (3, 6) match with peaks of V (2, 14), (4, 14), (1, 7), (3, 4) respectively. However, given the query Q modeled as [(1, 12), (1, 5), (3, 3)], it can be seen that, using the string matching method proposed by Wai and Chen, the trajectory V is not relevant to the query trajectory Q.

When the query contains multiple motion trajectories, Wei and Chen stored all the query trajectories in the same finite automata.[9] However, this considers neither the spatial nor the temporal relationships between the query motion trajectories. In VideoQ, Chang et al.[2] suggested imposing spatial/temporal constraints on the query result by using the idea of 2D string. 2D string preserves the spatial knowledge of a symbolic picture.[3] Based on the 2D string approach, Shearer et al. proposed the spatial indexing for video sequences and defined three types of video matching.[8] However, in this approach, either the video sequence is qualified or unqualified to the query sequence. The response to the query only returns the qualified video sequences, not relevant video sequences along with degrees of relevance.

3. RETRIEVAL VIA SINGLE MOTION TRAJECTORY

In the proposed approach, the motion trajectory is modeled as a sequence of segments, instead of peaks. Each segment is represented by the angle. The angle is defined as the slope of the segment. For example, the video motion trajectory V in Figure 1 is represented as (310°, 0°, 310°, 240°, 5°, 95°, 45°, 0°). The query motion trajectory P is represented as (0°, 270°, 260°, 0°, 90°) while Q is represented as (0°, 265°, 5°, 70°). Moreover, there exist two values for the difference, $d(v_i, q_j)$, between two angles of segments v_i, q_j. For example, if v_i equals 30° while q_j equals -30°, then $d(v_i, q_j)$ equals either 60° or 300°. Therefore, the difference, $d(v_i, q_j)$, is defined as follows.

If $|v_i - q_j| > 180°$ **then** $d(v_i, q_j) = (360° - |v_i - q_j|)$
else $d(v_i, q_j) = |(v_i - q_j)|$.

We first define the quantitative substring matching Optimal Consecutive Mapping (OCM) in the following. The algorithm is listed in Algorithm OCM.

Definition 1 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$, the video shot $V = (v_1, v_2, \ldots, v_N)$, $M \leqq N$, a consecutive mapping between them is a one-to-one relation R_{CM} from $\{1, 2, \ldots, M\}$ to $\{1, 2, \ldots, N\}$, such that

(1) for each i, $1 \leqq i \leqq M$, there exists one j, $1 \leqq j \leqq N$, such that $(i, j) \in R_{CM}$.
(2) for any two ordered pairs (i, j), (k, l) in R_{CM}, $[(j - l) = 1]$ if and only if $[(i - k) = 1]$.

Definition 2 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$, the video shot $V = (v_1, v_2, \ldots, v_N)$, and $d(q_i, v_j)$, $\forall\, i$, $1 \leqq i \leqq M$, $\forall\, j$, $1 \leqq j \leqq N$, the distance between Q and V for a given consecutive mapping R_{CM}, $D'_{RCM}(Q, V)$, is defined as

$$D_{R_{CM}}(Q,V) = \sum_{\forall (i,j) \in R_{CM}} d(q_i, v_j).$$

Definition 3 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$ and the video shot $V = (v_1, v_2, \ldots, v_N)$, the distance between Q and V for *Optimal Consecutive Mapping* is defined as

$$D_{OCM}(Q,V) = \min_{\forall R_{CM}} \{D_{R_{CM}}(Q,V)\}.$$

Algorithm Optimal Consecutive Mapping (OCM)
for $j = 0$ **to** $N\text{-}M$ **do** $D[0, j] = 0$;
for $j = 0$ **to** $N\text{-}M$ **do**
  for $i = 1$ **to** $M\text{-}1$ **do**
    $D[i, i+j] = D[i-1, i+j-1] + d(q_i, v_{i+j})$;
$D[M, M] = D[M-1, M-1] + d(q_M, v_M)$;
for $j = M+1$ **to** N **do**
    $D[M, j] = \min(D[M-1, j-1] + d(q_M, v_j), D[M, j-1])$;
return $D[M, N]$

Given the example in Figure 1, the computation of the quantitative similarity OCM between the video trajectory V and query trajectory P is listed in Table 2. Figure 3 describes the mapping relation between segments of V and P.

Table 2. Computation of OCM between motion trajectories V and P.

		310	*0*	*310*	*240*	*5*	*95*	*45*	*0*
	0	**0**	0	0					
0		50	**0**	50	120				
270			140	**40**	80	215			
260				190	**60**	185	380		
370					320	**75**	290	435	
90						405	**80**	**80**	**80**

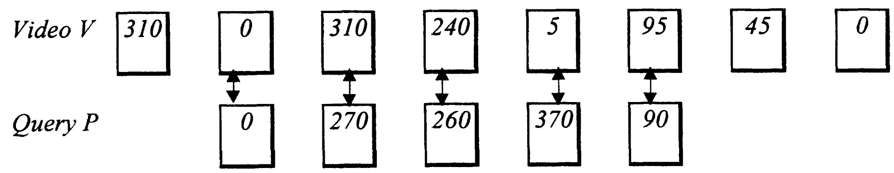

Figure 3. Mapping between segments of V and P using OCM.

OCM measures the similarity for one-to-one segment mapping between trajectories. However, in Figure 1, given query trajectory Q, represented as (0°, 265°, 5°, 70°), video trajectory V is not relevant using OCM. The reason comes from that intuitively the segment q_2 of Q is mapped with the segments v_3 and v_4 of V. Therefore, we extend the definition of Optimal Consecutive Mapping to Optimal Consecutive Mapping with Replication (OCMR). In OCMR, each segment of query trajectory is permitted to map with more than one segments of video trajectory. In addition, the same as OCM, the mapped segments must be consecutive.

Definition 4 Given the query video shot $Q = (q_1, q_2, \ldots, q_M)$, the video shot $V = (v_1, v_2, \ldots, v_N)$, $M \leqq N$, a consecutive mapping with replication between them is a one-to-many relation R_{CMR} from $\{1, 2, \ldots, M\}$ to $\{1, 2, \ldots, N\}$, such that

(1) for each i, $1 \leqq i \leqq M$, there exists at least one j, $1 \leqq j \leqq N$, such that $(i, j) \in R_{CMR}$,

(2) let $p_{max}(i)$ be $\max\{j \mid (i,j) \in R_{CMR}\}$, $p_{min}(i)$ be $\min\{j \mid (i,j) \in R_{CMR}\}$, for each j, $p_{min}(1) \leqq j \leqq p_{max}(M)$, there exists one i, $1 \leqq j \leqq M$, such that $(i, j) \in R_{CMR}$

(3) for any two ordered pairs (i, j), (k, l) in R_{CMR}, if $(i < k)$ then $(j < l)$,

(4) any two ordered pairs (i, j), (k, l) in R_{CMR}, if $[(j - l) = 1]$ then either $[(i - k) = 1]$ or $[(i - k) = 0]$.

Definition 5 Given the query video shot $Q = (q_1, q_2, \ldots, q_M)$, the video shot $V = (v_1, v_2, \ldots, v_N)$ and $d(q_i, v_j)$, $\forall\, i$, $1 \leqq i \leqq M$, $\forall\, j$, $1 \leqq j \leqq N$, the distance between Q and V for a given consecutive mapping R_{CMR}, $D'_{RCMR}(Q, V)$, is defined as

$$D_{R_{CMR}}(Q,V) = \sum_{\forall (i,j) \in R_{CMR}} d(q_i, v_j).$$

Definition 6 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$ and the video shot $V = (v_1, v_2, \ldots, v_N)$, the distance between Q and V for *Optimal Consecutive Mapping with Replication* is defined as

$$D_{OCMR}(Q,V) = \min_{\forall R_{CMR}} \{D_{R_{CMR}}(Q,V)\}.$$

The solution of OCMR can be obtained as follows. First, if a mapping R is an OCMR, then the first segment q_1 must be mapped with only one segment of video sequence, so does the last segment q_M. Otherwise, suppose that q_1 is mapped with segments $v_a, \ldots v_{i-1}, v_i$, q_M is mapped with $v_j, v_{j-1} \ldots, v_b$. We can derive a mapping with less distance by removing the mapping pairs between q_1 and $v_a, \ldots v_{i-1}$, and those between q_M and $v_{j-1} \ldots, v_b$. Therefore the behaviors of q_1 and q_M are the same as that of segment in OCM.

Next, without loss of generality, we assume that q_1 is mapped with v_a, q_M is mapped with v_b. Let $D[i, j]$ be the minimum cost of mapping with replication between $(v_{a+1}, v_{i+2}, \ldots, v_{b-1})$ and $(q_2, q_3, \ldots, q_{M-1})$. There are two possible relations between $D[m, n]$ and $D[i, j]$ for some combinations of smaller is and js.:

map: the segment v_n is mapped with the segment u_m, $D[m, n] = D[m\text{-}1, n\text{-}1] + d(u_m, v_n)$.

replicate v_n: the segment v_n is replicated to mapped with the segment u_m, $D[m, n] = D[m\text{-}1, n] + d(u_m, v_n)$.

Figure 4 illustrates the relation. Combining these two cases, we get the following recurrence relation.

$$D[m,n] = \min \begin{cases} D[m-1, n-1] + d(u_m, v_n) \\ D[m-1, n] + d(u_m, v_n) \end{cases},$$

with $D[0, 0] = 0$, $D[0, j] = 0$, for all j, $1 \leqq j \leqq N\text{-}M$, and $D[i, i\text{-}1] = \infty$, for all i, $1 \leqq i \leqq M$.

Algorithm OCMR lists the procedure.

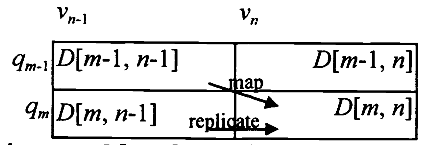

Figure 4. Relation of OCMR between $D[m, n]$ and $D[i, j]$ for some combinations of smaller is and js.

Algorithm Optimal Consecutive Mapping with Replication (OCMR)

```
for j = 0 to N-M do D[0, j] = 0;
For i = 1 to M-1 do D[i, i-1] = ∞;
for j = 0 to N-M do
    for i = 1 to M-1 do
        D[i, i+j] = min(D[i-1, i+j-1]+d(q_i, v_{i+j}), D[i, i+j-1]+d(q_i, v_{i+j}) );
D[M, M] = D[M-1, M-1]+d(q_M, v_M);
for j = M+ 1 to N do
    D[M, j] = min(D[M-1, j-1]+d(q_M, v_j), D[M, j-1]);
return D[M, N]
```

Table 3. Computation of OCMR between motion trajectories V and Q.

		310	*0*	*310*	*240*	*5*	*95*	*45*	*0*
	0	**0**	0	0	0				
0	∞	**50**	**0**	50	120	5			
265		∞	145	**45**	**70**	170	175		
5			∞	200	170	**70**	160	200	
70					370	235	**95**	**95**	**95**

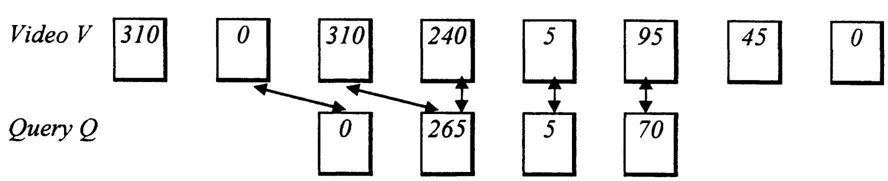

Figure 5. Mapping between segments of V and Q using OCMR.

Using the similarity OCMR, Table 3 describes the computation of the similarity between the video trajectory V and query trajectory Q. In Figure 5, it can be seen that the segment q_2 is mapped with v_3 and v_4.

4. RETRIEVAL VIA MULTIPLE MOTION TRAJECTORIES

When the query contains multiple motion trajectories, the result is simply an intersection of results of individual query motion trajectory. However, this simple approach considers neither the spatial nor the temporal relationships between the query motion trajectories. For example, a video shot, in which the policeman runs after a car, consists of two motion trajectories shown in Figure 6. From these two trajectories, no information is available about the spatial/temporal relationships indicating the action of chasing.

To impose the spatial/temporal constraint, multiple trajectories can be modeled as a sequence of symbolic pictures. Figure 7 shows a sequence of symbolic pictures. In these pictures, the symbols "C", "P" and "T" denote the car, the policeman and the tree respectively. From this sequence of pictures, the action of chasing can be derived. Spatial information of the symbolic picture can be represented by the spatial data structure, 2D string.[3] For the sequence of symbolic pictures, 2D strings can be extended to a sequence of 2D strings or a 2D string followed by a sequence of change edits.[1]

Shearer et al. defined three types of video matching.[8] Type-2 video matching displays a picture for picture equivalence between the two compared video sequences. Picture matching is the same as the original type-0, 1, 2 subpicture matching.

Type-1 video matching relaxes type-2 video matching by allowing repetitions of matching pictures. Type-0 video matching relaxes type-1 matching that the matching pictures need not to be consecutive pictures. Figure 8 describes these three types of video matching.

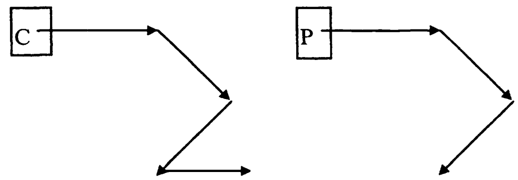

Figure 6. Examples of motion trajectories of a car and a policeman.

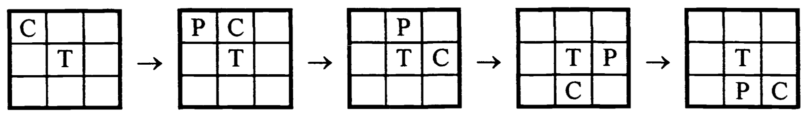

Figure 7. A sequence of symbolic pictures describing the chasing.

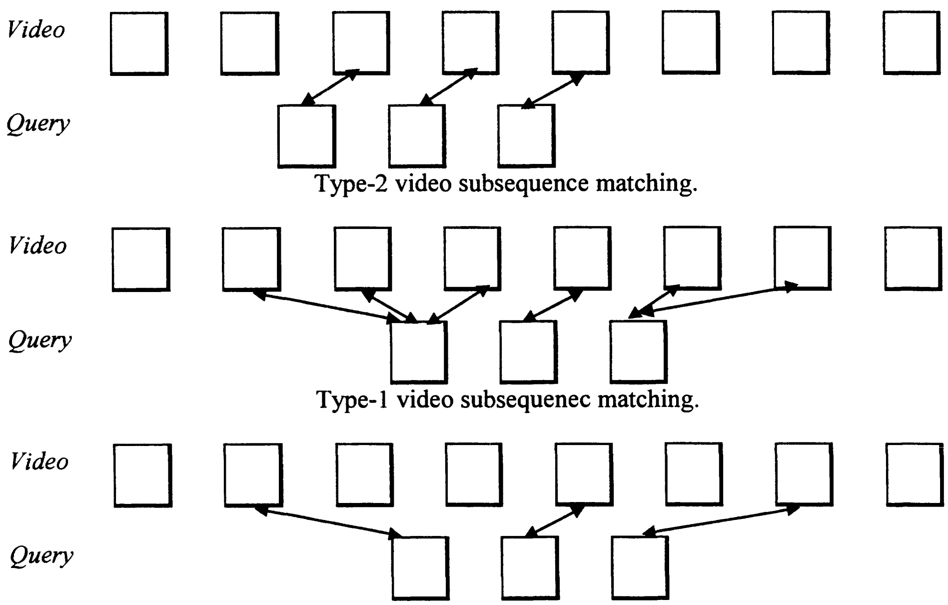

Figure 8. Three types of video subsequence matching.[6]

However, in this approach, either the video sequence is qualified or unqualified to the query sequence. The response to the query only return the qualified video sequences, not relevant video sequences along with degrees of relevance. We propose the similarity algorithms to measure the similarity between sequences of symbolic pictures, as long as the distance between symbolic pictures are available. A few approaches for computation of similarity (or distance) between symbolic pictures have been studied.[6, 7]

In the previous section, we have described the definitions and algorithms OCM and OCMR. It can be seen that the definition of OCM is the quantitative similarity matching corresponding to type-2 subsequence matching. Definition of OCMR is somewhat correspondent with type-1 video matching. Both OCMR and type-1 matching allow repetitions of matching frames with the exception that, in OCMR, only one image frame is allowed to mapped for the first and the last image frame. For example, given the distance between the symbolic pictures of video sequence *V* and *Q* (Table 4), the computation of type-2, type-1 video similarity between *V* and *Q* are presented in Table 5 and Table 6. Figure 9 and Figure 10 show the mapping for type-2 and type-1 similarity respectively.

Table 4. Distance between symbolic pictures of Q and V.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
q_1	0.3	0.1	0.2	0.4	0.1	0.5	0.1
q_2	0.1	0.3	0.4	0.2	0.3	0.7	0.1
q_3	0.6	0.2	0.1	0.7	0.2	0.2	0.4
q_4	0.2	0.2	0.3	0.3	0.2	0.6	0

Table 5. Computation of type-2 video similarity between Q and V.

		v_1	v_2	v_3	v_4	v_5	v_6	v_7
	0	0	0	**0**				
q_1		0.3	0.1	0.2	**0.4**			
q_2			0.6	0.5	0.4	**0.7**		
q_3				0.7	1.2	0.6	**0.9**	
q_4					1.0	1.0	1.0	**0.9**

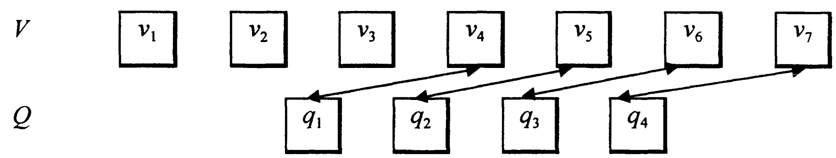

Figure 9. Mapping of type-2 video similarity between Q and V.

Table 6. Computation of Type-1 video similarity between Q and V.

		v_1	v_2	v_3	v_4	v_5	v_6	v_7
	0	**0**	**0**	0				
q_1	0	0.3	0.1	**0.2**	0.4			
q_2		∞	0.6	0.5	**0.4**	0.7		
q_3			∞	0.7	1.2	**0.6**	**0.8**	0.5
q_4					1.0	1.0	1.0	**0.8**

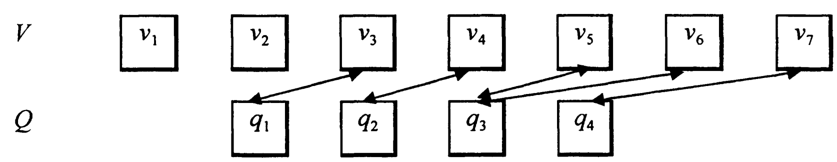

Figure 10. Mapping of type-1 video similarity between Q and V.

For the quantitative similarity matching corresponding to type-0 subsequence matching, we define Optimal Subsequence Mapping (OSM) and present the corresponding algorithm as follows.

Definition 7 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$ and the video shot $V = (v_1, v_2, \ldots, v_N)$, $M \leqq N$, a subsequence mapping between them is a one-to-one relation R_{SM} from $\{1, 2, \ldots, M\}$ to $\{1,2, \ldots, N\}$, such that

(1) for each i, $1 \leqq i \leqq M$, there exists one j, $1 \leqq j \leqq N$, such that $(i, j) \in R_{SM}$.

(2) for any two ordered pairs (i, j), (k, l) in R_{SM}, $(j < l)$ if and only if $(i < k)$.

Definition 8 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$, the video shot $V = (v_1, v_2, \ldots, v_N)$, and the distance $d(q_i, v_j)$, $\forall\, i$, $1 \leqq i \leqq M$, $\forall\, j$, $1 \leqq j \leqq N$, the distance between Q and V for a given subsequence mapping R_{SM}, $D'_{RSM}(Q, V)$, is defined as

$$D_{R_{SM}}(Q,V) = \sum_{\forall (i,j) \in R_{SM}} d(q_i, v_j).$$

Definition 9 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$ and the video shot $V = (v_1, v_2, \ldots, v_N)$, the distance between Q and V for *Optimal Subsequence Mapping* is defined as

$$D_{OSM}(Q,V) = \min_{\forall R_{SM}} \{D_{R_{SM}}(Q,V)\}.$$

Algorithm <u>Optimal Subsequence Mapping (OSM)</u>
for $i = 1$ **to** M **do** $D[i, i\text{-}1] = \infty$;
for $j = 0$ **to** $N\text{-}M$ **do** $D[0, j] = 0$;
for $i = 1$ **to** M **do**
    for $j = i$ **to** $i + N\text{-}M$ **do**
        $D[i, j]=\min(D[i\text{-}1, j\text{-}1]+d(q_i, v_j), D[i, j\text{-}1])$;
return $D[M, N]$

Table 7. Computation of type-0 video similarity between Q and V.

		v_1	v_2	v_3	v_4	v_5	v_6	v_7
	0	**0**	0	0				
q_1	∞	0.3	**0.1**	**0.1**	0.1			
q_2		∞	0.6	0.5	**0.3**	0.3		
q_3			∞	0.7	0.7	**0.5**	**0.5**	
q_4				∞	1.0	0.9	0.9	**0.5**

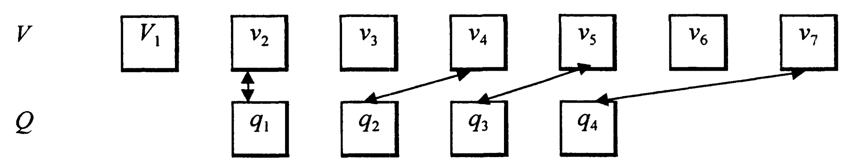

Figure 11. Mapping of type-0 video similarity between Q and V.

Table 7 is the computation of type-0 video similarity measure. The mapping result is shown in Figure 11. Sometimes, it is possible that the trajectory of query Q is very similar to that of video V, except that few pictures are very dissimilar. Using the similarity measures mentioned previously, these dissimilar pairs of pictures produce large distance. Therefore, in the following definitions, the mapping is constrained by a threshold δ.

Definition 10 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$, the video shot $V = (v_1, v_2, \ldots, v_N)$, the frame distance tolerance δ and the distance $d(q_i, v_j)$, $\forall\ i$, $1 \leqq i \leqq M$, $\forall\ j$, $1 \leqq j \leqq N$, a distance-constrained subsequence mapping between them is a one-to-one relation R_{DSM} from $\{1, 2, \ldots, M\}$ to $\{1, 2, \ldots, N\}$, such that
(1) for each order pair (i, j) in R_{DSM}, $d(u_i, v_j) \leqq \delta$,
(2) for each i, $1 \leqq i \leqq M$, there exists one j, $1 \leqq j \leqq N$, such that $(i, j) \in R_{DSM}$,
(3) for any two ordered pairs (i, j), (k, l) in R_{DSM}, $(j < l)$ if and only if $(i < k)$.

Definition 11 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$, the video shot $V = (v_1, v_2, \ldots, v_N)$, the frame distance constraint δ, and the distance $d(q_i, v_j)$, $\forall\ i$, $1 \leqq i \leqq M$, $\forall\ j$, $1 \leqq j \leqq N$, the distance between Q and V for a given mapping R_{DSM}, $D'_{R_{DSM}}(Q, V, \delta)$, is defined as

$$D_{R_{DSM}}(Q,V,\delta) = \sum_{\forall(i,j)\in R_{DSM}} d(q_i, v_j).$$

Definition 12 Given the query shot $Q = (q_1, q_2, \ldots, q_M)$ and the video shot $V = (v_1, v_2, \ldots, v_N)$, the distance between Q and V for *Distance- constrained Optimal Subsequence Mapping(DOSM)* is defined as

$$D_{DOSM}(Q,V,\delta) = \min_{\forall R_{DSM}} \{D_{R_{DSM}}(Q,V,\delta)\}.$$

If there is no such mapping, the distance $D_{DOSM}(Q, V, \delta)$ is set to ∞.

Algorithm <u>Distance-constrained Optimal Subsequence Mapping (DOSM)</u>
for $i = 1$ **to** M **do** $D[i, i\text{-}1] = \infty$;
for $j = 0$ **to** $N\text{-}M$ **do** $D[0, j] = 0$;
for $i = 1$ **to** M **do**
    for $j = i$ **to** $i + N\text{-}M$ **do**
        if $d(q_i, v_j) < \delta$ **then**

$D[i,j]$=min($D[i-1,j-1] + d(q_i, v_j)$, $D[i,j-1]$)
else
$D[i,j]= D[i,j-1]$;
return $D[M, N]$

For example, given the distances in Table 4, the computation of video similarity between V and Q using DOSM are presented in Table 8. In this table, the distance constraint δ is set to 0.3. Figure 12 shows the mapping result.

Table 8. Computation of distance-constrained video similarity between Q and V.

		v_1	v_2	v_3	v_4	v_5	v_6	v_7
	0	**0**	0	0				
q_1	∞	∞	**0.1**	**0.1**	0.1			
q_2		∞	∞	∞	**0.3**	0.3		
q_3			∞	∞	∞	**0.5**	**0.5**	
q_4				∞	∞	∞	∞	**0.5**

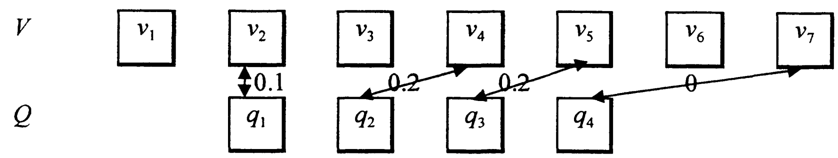

Figure 12. Mapping of distance-constraint video similarity between Q and V (δ=0.3).

5. NORMALIZATION OF SIMILAITY MEASURES

Concerning the normalization of similarity measures between sequence of frames, it is unfair for video with more number of frames to be measured by the algorithms allowing replications. For example, considering two video sequence V_1, V_2 and the query sequence Q shown in Figure 13, the distances $R_{OCMR}(Q, V_1)$ and $R_{OCMR}(Q, V_2)$ both equal 80. However, intuitively, V_1 is more similar to Q, though there are more number of mapping in V_1. To deal with this unfairness, we normalize the distance by the cardinality of relation R_{OCMR}. In this example, the normalized $R_{OCMR}(Q, V_1)$ becomes 80/6 while the normalized $R_{OCMR}(Q, V_2)$ becomes 80/2.

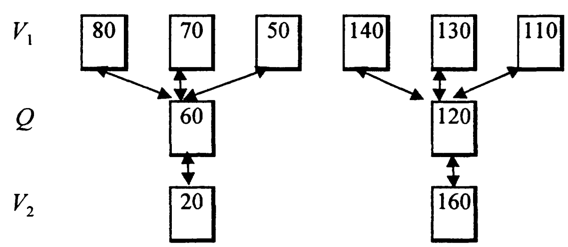

Figure 13. $R_{OCMR}(Q, V_1)$ and $R_{OCMR}(Q, V_2)$.

6. CONCLUSIONS

Content-based video retrieval is an important issue in applications of multimedia information systems. Motion trajectory is one of the most prominent features of video. Two distinguished features of motion trajectory matching are sub-matching and approximate matching.

In this paper, we propose quantitative similarity approaches for video retrieval via single motion trajectory and multiple motion trajectories. For the retrieval via single motion trajectory, the trajectory is modeled as a sequence of segments and each segment is represented as the slope. Two quantitative similarity measures and corresponding algorithms based on the sequence similarity are presented. For the retrieval via multiple motion trajectories, the trajectories of the video are modeled as a sequence of symbolic pictures. Four quantitative similarity measures and algorithms, which are also based on the sequence similarity, are proposed. Evaluating the performance of the proposed similarity measures is left as future work.

7. REFERENCE

1. T. Arndt and S. K. Chang, “Image Sequence Compression By Iconic Indexing,” In *Proceedings of IEEE Workshop on Visual Languages*, Los Alamitos, CA, pp. 177-182, 1989.
2. S. F. Chang, W. Chen, H. J. Meng, H. Sundaram and D. Zhong, “VideoQ: An Automated Content Based Video Search System Using Visual Cues,” In *Proceedings of ACM Multimedia'97*, Seattle, WA, pp. 313-324, 1997.
3. S. K. Chang, Q. Y. Shi and C. W. Yang C. W., “Iconic Indexing by 2-D Strings,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 3, pp. 413-428, 1987.
4. N. Dimitrova and F. Golshani, “Rx for Semantic Video Database Retrieval,” In *Proceedings of ACM Multimedia'94*, San Francisco, CA, pp. 219-226, 1994.
5. N. Dimitrova and F. Golshani, “Motion Recovery for Video Content Classification,” *ACM Transactions on Information Systems*, Vol. 13, No. 4, pp. 408-439, 1995.
6. V. N. Gudivada and V. V. Raghavan, “Design and Evaluation of algorithms for Image retrieval by Spatial Similarity,” *ACM Transactions on Information Systems*, Vol. 13, No. 2, pp. 114-144, 1995.
7. Suh-Yin Lee, Man-Kwan Shan and Wei-Pang Yang, “Similarity Retrieval of Iconic Image Database,” *Pattern Recognition*, Vol. 22, No. 6, pp. 675-682, 1989.
8. K. Shearer, S. Venkatesh and D. Kieronska, “Spatial Indexing for Video Databases,” *Journal of Visual Communication and Image Representation*, Vol. 7, No. 4, pp. 325-335, 1996.
9. T. T. Y. Wai and A. L. P. Chen, “Retrieving Video Data via Motion Tracks of Content Symbols,” In *Proceedings of ACM International Conference on Information and Knowledge Management CIKM'97*, Las Vegas, NV, pp. 105-112, 1997.
10. A. Yoshitaka, M. Yoshimitsu, M. Hirakawa and T. Ichikawa, “V-QBE: Video Database Retrieval by Means of Example Motion of Objects,” in *Proceedings of IEEE InternationalConference on Multimedia Computing and Systems'96*, Hiroshima, Japan, pp. 453-457, 1996.