

# 비디오 데이터베이스에서 이동 객체의 유사 부분 움직임 궤적을 위한 N-워핑 검색

심춘보<sup>o</sup>      장재우  
전북대학교 컴퓨터공학과  
{cbsim, jwchang}@dblab.chonbuk.ac.kr

## N-Warping Searches for Similar Sub-Trajectories of Moving Objects in Video Databases

Choon-Bo Shim<sup>o</sup>      Jae-Woo Chang

Dept. of Computer Engineering, Chonbuk National University

### 요약

본 논문에서는 비디오 데이터가 지나는 이동 객체의 움직임 궤적(moving objects' trajectories)에 대해 유사 부분 움직임 궤적 검색을 효율적으로 지원하는 N-워핑(N-warping) 알고리즘을 제안한다. 제안하는 알고리즘은 기존의 시계열 데이터베이스에서 유사 서브시퀀스 검색을 위해 사용되었던 타임 워핑 변환 기법(time-warping transformation)을 변형한 알고리즘이다. 또한 제안하는 알고리즘은 움직임 궤적을 모델링하기 위해 사용되는 단일 속성(property)인 각도뿐만 아니라, 거리와 시간과 같은 다중 속성을 지원하며, 사용자 질의에 대해 유사 부분 움직임 궤적 검색을 가능하게 하는 근사 매칭(approximate matching)을 지원한다.

### 1. 서론

시간의 흐름에 따라 공간적 위치가 연속적으로 변하는 객체를 이동객체(moving objects)라 하며, 이러한 이동 객체의 연속적인 움직임들의 모임을 움직임 궤적(trajectories)이라 한다[1][2]. 이러한 움직임 궤적은 시공간 데이터베이스나 비디오 데이터베이스에서 사용자의 주된 관심의 대상이며, 내용 기반 검색을 수행하는 데 있어 매우 중요한 역할을 수행한다. 그리고, 주어진 사용자 질의 움직임 궤적과 유사한 움직임 궤적을 포함하는 이동 객체의 움직임 궤적을 찾는 것을 유사 부분 움직임 궤적 검색(similar sub-trajectory retrieval)이라 한다. 유사 부분 움직임 궤적 검색을 지원하기 위해서는 질의 움직임 궤적과 주어진 어느 정도의 허용치 범위 내에서 유사한 데이터 움직임 궤적을 검색할 수 있는 근사 매칭을 지원해야 한다.

이를 위해, 본 논문에서는 비디오 데이터가 지나는 이동 객체의 움직임 궤적(moving objects' trajectories)에 대해 유사 부분 움직임 궤적 검색을 효율적으로 지원하는 N-워핑(N-warping) 알고리즘을 제안한다. 제안하는 알고리즘은 기존의 시계열 데이터베이스에서 유사 서브시퀀스 검색을 위해 사용되었던 타임 워핑 변환 기법(time-warping transformation)을 변형한 알고리즘이다. 또한 제안하는 알고리즘은 움직임 궤적을 모델링하기 위해 사용되는 단일 속성(property)인 각도뿐만 아니라, 거리와 시간과 같은 다중 속성을 지원하며, 사용자 질의에 대해 유사한 부분 움직임 궤적 검색을 가능하게 하는 근사 매칭을 효율적으로 지원한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고, 3장에서는 이동 객체의 유사 부분 움직임 궤적을 지원하는 N-워핑 검색 알고리즘을 제안한다. 4장에서는 유사 부분 움직임 궤적을 위한 N-워핑 검색의 예를 보이고, 마지막으로, 5장에서는 결론 및 향후 연구 방향을 제시한다.

### 2. 관련 연구

시퀀스 데이터베이스[3][4][5]는 다양한 길이의 시퀀스들로 구성되며 시퀀스  $S(=s[1], s[2], \dots, s[|S|])$ 는 일정한 시간 주기마다 얻어진 연속된 실수 값들로 이루어진다. 여기서,  $|S|$ 는 시퀀스의 길이이다.  $s[i]$ 는  $S$ 의  $i$ 번째 요소를 나타내

며,  $s[i:j]$ 는  $i$ 번째 위치에서  $j$ 번째 위치를 포함하는 서브시퀀스를 의미한다.  $s[i:-]$ 는  $i$ 번째 위치에서 시작해서 시퀀스  $S$ 의 마지막 요소까지를 나타낸다.  $()$ 는 요소가 없는 널 시퀀스(null sequence)를 의미한다.

정의 1: 길이가  $n$ 인 두 시퀀스  $S$ 와  $Q$  사이의 유사한 정도를 계산하기 위한 거리 함수  $L_p$ 는 다음과 같이 정의된다.  $L_1$ 는 맨하탄 거리(manhattan distance),  $L_2$ 는 유클리드 거리(euclidean distance),  $L_\infty$ 는 대응되는 각 쌍의 거리 중 최대 거리를 의미한다.

$$L_p(S, Q) = \left( \sum_{i=1}^n |s[i] - q[i]|^p \right)^{\frac{1}{p}}, 1 \leq p < \infty$$

효율적인 유사 서브시퀀스 검색을 위해 시퀀스내의 각 요소 값을 임의의 수만큼 반복시키는 것을 허용하는 타임 워핑 거리를 이용한 유사 서브시퀀스 검색 기법이 제안되었다[3][4][5].

정의 2: 두 시퀀스  $S$ 와  $Q$  사이의 타임 워핑 거리  $D_w$ 는 다음과 같이 재귀적으로 정의된다.  $D_{base}$ 는 기본 거리 함수로서  $L_p$  중 응용에 적합한 것을 선택하여 사용한다.

$$\begin{aligned} D_w((), ()) &= 0 \\ D_w(S, ()) &= D_w((), Q) = \infty \\ D_w(S, Q) &= D_{base}(S[1], Q[1]) + \\ &\min(D_w(S, Q[2:-]), D_w(S[2:-], Q), D_w(S[2:-], Q[2:-])) \\ D_{base}(a, b) &= |a - b| \end{aligned}$$

Shan[2]은 움직임 궤적을 단일 속성인 각도(0~360)로 표현하며, 사용자 질의 움직임 궤적  $Q=(q_1, q_2, \dots, q_n)$ 와 유사한 움직임 궤적을 포함하는 데이터 움직임 궤적  $V=(v_1, v_2, \dots, v_n)$ 사이의 유사성을 측정하기 위해 OCM(Optimal Consecutive Mapping)과 OCMR(Optimal Consecutive Mapping with Replication)이라는 두 가지 유사성 측정 알고리즘을 제안하였다. 그 중에서 OCMR 알고리즘은 근사 매칭을 지원한다. 이를 위해, 사용자로부터 주어진 질의 움직임 궤적을 이루는 각각의 움직임 요소(Motion)들( $q_i$ )과 데이터 움직임 궤적을 이루는 각각의 움직임 요소들( $v_j$ ) 사이에 유사성을 계산하는

데 있어, 질의 움직임 요소들 중에서 자기 자신을 반복시킨 것(replicate)과 그렇지 않은 것(map)과 비교해서 더 작은 것을 선택한다. 여기서  $d(q_i, v_j)$ 는  $q_i$ 와  $v_j$ 사이의 거리 함수(distance function)이고  $D[M,N]$ 는 질의 움직임 궤적과 데이터 움직임 궤적사이의 반복(replication)을 이용한 최소 거리(minimum distance)를 구하기 위한 테이블이다. 그림 1은 OCMR알고리즘을 나타낸다.

```

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(qi, vi+j),
                  D[i, i+j-1]+d(qi, vi+j));
D[M,M]=D[M-1,M-1]+d(qM, vM);
for j=M+1 to N do
  D[M, j]=min(D[M-1, j-1]+d(qM, vj), D[M, j-1]);
return D[M,N];
    
```

그림 1 OCMR 알고리즘

### 3. 유사 부분 움직임 궤적을 위한 N-위평 검색

본 논문에서는 비디오 데이터베이스에서 효율적인 유사 부분 움직임 궤적 검색을 위한 근사 매칭을 위해 다음의 세 가지 요구사항(requirement)을 제시한다.

요구사항 1 : 유사 시퀀스 검색에서 사용되었던 타임 워핑 변환은 질의 시퀀스 뿐만 아니라 데이터 시퀀스에 대해서도 임의의 요소가 무한히 반복을 허용한다. 그러나 유사 움직임 궤적 검색에서는 단지 질의 움직임 궤적을 구성하는 움직임 요소들에 대해서만 반복을 허용한다.

요구사항 2 : 타임 워핑 변환은 임의의 요소들을 무한히 반복하는 것을 허용하는 것에 비해 N-위평 변환은 임의의 요소를 N번까지만 반복을 허용한다.

요구사항 3 : 움직임 궤적을 구성하는 움직임 요소는 단일 속성(각도) 뿐만 아니라, 다중 속성(거리, 시간 등)을 지원해야 한다.

여기서, 요구사항 1은 일반적으로 유사 부분 움직임 궤적 검색에서 근사 매칭을 지원하기 위함이고, 요구사항 2와 3은 요구사항 1을 이용한 근사 매칭의 효율성을 향상시키기 위한 것이다. 또한 요구사항 2와 3은 유사 부분 움직임 궤적 검색을 적용하려는 응용 분야(축구나 하키와 같은 비디오 데이터 검색 분야)에 따라 매우 중요한 사항이다.

기존의 타임 워핑을 이용한 유사 부분 시퀀스 검색 기법은 위의 세 가지 요구사항 모두를 만족시키지 않는다. 이는 시퀀스 데이터베이스에서 사용하는 연속된 실수 값들로 이루어진 시퀀스 데이터의 특성과 비디오 데이터베이스에서 이동 객체의 움직임 궤적 데이터 특성이 서로 상이하기 때문이다. 아울러, 기존의 근사 매칭을 위한 Shan의 OCMR 알고리즘은 요구사항 1은 만족하지만, 요구사항 2와 3은 만족시키지 못한다.

따라서, 위의 세 가지 요구사항을 모두 만족시키는 유사 부분 움직임 궤적 검색을 지원하기 위해, 본 논문에서는 이동 객체의 움직임 궤적(moving objects' trajectories)을 이용한 사용자 질의에 대해 효율적인 유사 부분 움직임 궤적 검색과 더불어 근사 매칭을 지원할 수 있는 새로운 N-위평 검색 알고리즘을 제안한다. 제안하는 알고리즘은 기존의 시계열 데이터베이스에서 유사 서브시퀀스 검색을 위해 사용되었던 타임 워핑 변환 기법을 변형한 N-위평 변환 기법을 이용한다.

본 논문에서는 요구사항 3을 만족시키기 위해, n차원의 속성으로 구성된 움직임 요소를 가지는 움직임 궤적을 정의 3과

같이 표현한다.

정의 3 : 움직임 궤적  $S(= \langle s[1], s[2], \dots, s[|S|] \rangle)$ 는 일련의 연속된 움직임 요소들의 집합으로 표현하며, 움직임 궤적을 구성하는 임의의 움직임 요소  $s[i]=\langle s[i,1], s[i,2], \dots, s[i,n] \rangle$ 로 나타낸다.

한편, 본 논문에서는 2차원의 속성을 지닌 움직임 궤적을 사용하며, 이는 정의 4와 같이 표현한다.

정의 4 : 움직임 궤적  $S(= \langle s[1], s[2], \dots, s[|S|] \rangle)$ 는 일련의 연속된 움직임 요소들의 집합으로 표현하며, 임의의 움직임 요소들은 각각 각도(angle)와 거리(distance)로 표현한다. 즉, 움직임 궤적을 구성하는 임의의 움직임 요소  $s[i]=\langle A_i, D_i \rangle$ 로 나타낸다.

주어진 질의 움직임 궤적의 임의의 움직임 요소와 데이터 움직임 궤적의 임의의 움직임 요소 사이의 유사성을 측정하기 위해, 본 논문에서는 정의 5와 같은 거리 함수  $d_{dr}$ 를 정의한다.

정의 5 : 움직임 궤적 S를 구성하는 임의의 움직임 요소  $s[i]$ 와 움직임 궤적 Q를 구성하는 임의의 움직임 요소  $q[j]$ 사이의 거리 함수(distance function)  $d_{dr}(s[i], q[j])$ 는 다음과 같이 정의된다. 여기서  $d_{ang}$ 는 움직임 요소의 각도간의 거리 함수를 나타내고,  $d_{dis}$ 는 움직임 요소의 거리간의 거리 함수를 의미한다. 그리고  $s[i,1]$ 와  $s[i,2]$ 는 각각 움직임 궤적 S의 i 번째 움직임 요소를 구성하는 각도와 거리를 나타낸다.  $\alpha$ 와  $\beta$ 는 각각 각도와 거리를 위한 가중치를 나타낸다. ( $\alpha + \beta = 1.0$ )

$$\begin{aligned}
 & \text{if } |s[i,1] - q[j,1]| > 180 \text{ Then} \\
 & \quad d_{ang}(s[i,1], q[j,1]) = (360 - |s[i,1] - q[j,1]|) \\
 & \text{else} \\
 & \quad d_{ang}(s[i,1], q[j,1]) = |s[i,1] - q[j,1]| \\
 & \quad d_{dis} = |s[i,2] - q[j,2]|
 \end{aligned}$$

$$d_{dr} = ((d_{ang}/180) * \alpha) + ((d_{dis}/100) * \beta)$$

요구사항 1과 2를 만족시키면서 효율적인 유사 부분 움직임 궤적 검색을 위해 기존의 유사 시퀀스 검색에서 사용되었던 정의 2를 변형하여 정의 6과 같은 새로운 N-위평 거리 함수를 정의한다.

정의 6 : 두 움직임 궤적 S와 Q 사이의 N-위평 거리  $D_{nw}$ 는 다음과 같이 재귀적으로 정의된다.

$$\begin{aligned}
 D_{nw}(\langle \rangle, \langle \rangle) &= 0 \\
 D_{nw}(S, \langle \rangle) &= D_{nw}(\langle \rangle, Q) = \infty \\
 D_{nw}(S, Q) &= D_{base}(S[1], Q[1]) + \\
 & \min(\{D_{nw}(S[2+k:-], Q), 0 \leq k < N\}, D_{nw}(S[2:-], Q[2:-])) \\
 D_{base}(a, b) &= d_{dr}(a, b)
 \end{aligned}$$

그림 2은 N-위평 알고리즘에 기반하여 주어진 질의 움직임 궤적과 유사한 부분 움직임 궤적을 포함하는 데이터 움직임 궤적을 검색하는 알고리즘이다. 여기서 Gen\_Sub\_Trajectory() 루틴은 데이터 움직임 궤적을 주어진 질의 움직임 궤적을 구성하는 움직임 요소의 수(=|Q|)와 반복 제한 횟수(N)을 이용하여 데이터 부분 움직임 궤적(data sub-trajectory)을 생성하는 함수이다. 그리고 NW\_CM\_Table()는 요구사항 1을 만족시키는 함수로 기존의 타임 워핑 거리 함수를 계산하기 위한 루틴을 변형하여 주어진 질의 움직임 궤적 Q를 구성하는 임의의 움직임 요소만을 무한히 반복해서 누적 테이블(nwTbl)을 만든다. 아울러, NW\_Distable()는 요구사항 2를 만족시키는 함수로 질의 움직임 궤적 Q를 구성하는 임의의 움직임 요소를 주어진 반복 횟수인 N번으로 제한해서 질의 움직임 궤적 Q와 데이터 움직임 궤적 S사이의 N-위평 거리를 계산하는 함수이다.

```

Algorithm Sub_Traj_Search(S, Q, N)
{
input :
S: Data Trajectory, Q: Query Trajectory;
N : the number of warping;
output :
minNWDist : minimum distance acquired using N-warping

for i=|Q| to |Q|*(N+1) do {
for j=0 to |S|-i do {
minNWDist=∞ ; nwTbl=[][]; ST=[];
Gen_Sub_Trajectory(S, i, j, ST);
NW_CM_Table(i, ST, |Q|, Q, nwTbl);
nwDist=NW_Distance(i, ST, |Q|, Q, nwTbl, N);
if nwDist < minNWDist
minNWDist=nwDist;
}
}
return minNWDist;
}

Function NW_CM_Table(S, Q)
{
input :
S: Data Trajectory, Q: Query Trajectory;
output :
nwTbl[][]: cumulative distance table;

nwTbl[0][0] = dr(S[0], Q[0]);
for j=1 to |Q|-1 do
nwTbl[j-1][j] = ∞ ;
for i=1 to |S|-1 do
nwTbl[i][0] = dr(S[i], Q[0]) + nwTbl[i-1][0];
for i=1 to |Q|-1 do
for j=i to |S|-1 do
nwTbl[i][j] = dr(S[i], Q[j]) +
min(nwTbl[i-1][j], nwTbl[i-1][j-1]);
}
}

Function NW_Distance(S, Q, nwTbl, N)
{
input :
S: Data Trajectory, Q: Query Trajectory;
nwTbl[][]: cumulative distance table;
N : the number of warping;
output :
minDist: minimum distance acquired using N-warping

lo_wp_cnt=N; //local warping count
gl_wp_cnt=|S|-|Q|; //global warping count
i=|Q|-1; j=|S|-1;
minDist = dr(S[j], Q[i]);
while(1) {
if( i==0 && j==0) break;
else {
if( (i>0) && (nwTbl[i][j-1] < nwTbl[i-1][j-1]) &&
lo_wp_cnt>0) {
j--; lo_wp_cnt--; gl_wp_cnt--;
minDist += dr(S[j], Q[i]);
}
else if( (i*N<gl_wp_cnt) && (lo_wp_cnt>0) &&
gl_wp_cnt!=0) {
j--; lo_wp_cnt--; gl_wp_cnt--;
minDist += dr(S[j], Q[i]);
}
else {
i--; j--; lo_wp_cnt = N;
minDist += dr(S[j], Q[i]);
}
}
}
return minDist;
}
    
```

그림 2 N-워핑을 이용한 유사 부분 움직임 궤적 검색 알고리즘

4. 유사 부분 움직임 궤적 검색의 예

그림 4는 본 논문에서 제안하는 N-워핑 알고리즘을 이용해 유사 부분 움직임 궤적을 검색한 예이다. 즉, 데이터 움직임 궤적 S에서 주어진 질의 움직임 궤적 Q와 유사한 부분 움직임 궤적을 매칭하는 것으로 기존의 Shan의 OCMR 알고리즘을 적용해서 가장 유사한 부분 움직임 궤적을 구하면 단일 점선(S<sub>1</sub>)으로 표시된 부분 즉, s[0]에서부터 s[5]까지가 가장 유사한 부분 움직임 궤적으로 매칭해서 0.87이라는 유사도(similarity)를 반환한다. 이는 OCMR 알고리즘이 무한히 반복하기 때문으로, q[1]의 경우 s[1], s[2], s[3] 그리고 s[4]와 반복해서 대응된다.

그에 비해, 제안하는 N-워핑 알고리즘을 이용하면 대시 점선(S<sub>2</sub>)으로 표시된 부분 즉, s[0]에서부터 s[3]까지를 가장 유사한 부분 움직임 궤적으로 매칭해서 0.93이라는 유사도를 반환한다. 이는 반복 제한 횟수 N(=2)만큼 반복을 하기 때문이며 또한 각도와 거리를 같이 고려하기 때문이다. 따라서, 제안하는 N-워핑 알고리즘이 Shan의 OCMR 알고리즘보다 사용자 질의에 대해 보다 유사하고 정확한 결과를 반환한다.

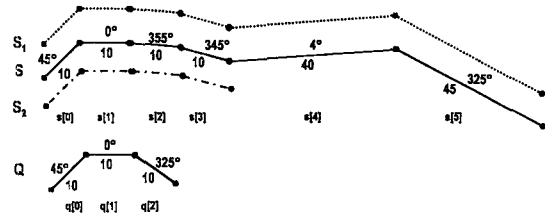


그림 4 유사 부분 움직임 궤적 검색의 예

5. 결론

본 논문에서는 비디오 데이터가 지니는 이동 객체의 움직임 궤적(moving objects' trajectories)에 대해 효율적인 유사 부분 움직임 궤적 검색과 더불어 근사 매칭을 지원할 수 있는 새로운 N-워핑 검색 알고리즘을 제안한다. 제안하는 알고리즘은 움직임 궤적을 모델링하기 위해 단일 속성(property)인 각도뿐만 아니라, 거리와 시간과 같은 다중 속성을 지원하며, 근사 매칭을 위해 제안되었던 Shan의 알고리즘보다 유사하고 정확한 부분 움직임 궤적을 검색할 수 있다.

향후 연구 과제는 제안하는 N-워핑 검색 알고리즘의 유용성을 검증하기 위해 실제 비디오 데이터로부터 추출한 대용량의 움직임 궤적 데이터를 이용해서 Shan 방법과 성능 평가를 수행하는 것이다.

참고문헌

- [1] T. T. Y. Wai and A. L. P. Chen, "Retrieving Video Data via Motion Tracks of Content Symbols," In. Proc. Int'l. ACM Conf. on Information and Knowledge Management CIKM'97, Las Vegas, NV, pp. 105-112, 1997
- [2] Man-Kwan Shan and Suh-Yin Lee, "Content-based Video Retrieval via Motion Trajectories", In Proceedings of the International Conference on SPIE, Vol. 3561, pp. 52-61, 1998
- [3] B. K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient Retrieval of Similar Time Sequences Under Time Warping," In. Proc. Int'l. Conf. on Data Engineering, IEEE, pp. 201-208, 1998
- [4] S. H. Park et al., "Efficient Searches for Similar Subsequence of Difference Lengths in Sequence Databases," In. Proc. Int'l. Conf. on Data Engineering, IEEE, pp. 23-32, 2000
- [5] S. W. Kim, S. H. Park, and W. W. Chu, "An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases," In. Proc. Int'l. Conf. on Data Engineering, IEEE, pp. 607-614, 2001