

움직임 객체의 궤적 유사도 검색

김미희, 복경수, 유재수, 조기형
충북대학교 정보통신공학과

Trajectory Similarity Retrieval of Moving Objects

Mi Hee Kim, Kyoung Soo Bok, Jae Soo Yoo, Ki Hyung Cho
Dept. of Computer & Communication Engineering, Chungbuk Nat'l University

요 약

최근 멀티미디어 데이터에 대한 관심이 높아지면서 비디오 데이터에 대한 연구가 매우 활발히 진행되고 있다. 비디오 데이터에는 시간의 변화에 따라 공간적인 위치가 변화하는 움직임 객체를 포함하고 있다. 이러한 움직임 객체를 효과적으로 표현하고 검색하기 위해서는 객체의 전체적인 움직임을 파악할 수 있는 궤적을 표현하고 검색할 수 있는 방법이 필요하다. 본 논문에서는 비디오 내의 움직임 객체의 궤적을 효과적으로 표현하고 검색하는 기법을 제안한다. 따라서 움직임 객체의 궤적 유사도 검색을 수행하도록 한다.

1. 서론

전세계적으로 인터넷과 초고속 정보 통신망의 구축 등으로 멀티미디어에 대한 관심이 집중되면서 멀티미디어 컴퓨팅, 통신 그리고 이에 대한 응용 기술들이 급속도로 발전하고 있다. 멀티미디어 데이터 중에서도 비디오 데이터는 대용량이고 비정형화인 데이터 특성 뿐만 아니라 시청각 정보, 시공간 정보, 객체와 사건과 같은 의미 정보 등을 포함하고 있다. 이러한 비디오 데이터를 효율적으로 저장하고 검색하기 위해서는 응용 목적에 따라 사용자의 요구와 질의 형태를 분석하여 적절히 모델링하는 작업과 다양한 사용자의 요구를 처리하기 위한 새로운 검색 기법이 필요하다.

비디오에 존재하는 움직임 객체는 시간의 변화에 따라 공간의 변화를 가지는 시공간적 특성을 가진다. 움직임 객체를 효과적으로 표현하기 위해서는 객체의 전체적인 움직인 모양을 파악할 수 있는 궤적(trajjectory)을 표현해야 한다. 그러나 기존의 연구들은 대부분 객체의 시간에 따른 공간적인 위치만 표현하므로써 공간적 크기 변화 또는 다른 객체와의 거리 변화를 고려하지 않고 있다[1,2,3,4]. 또한 움직임 객체

의 궤적을 표현하기 위해 일반적으로 방향만을 고려하고 있다는 문제점이 있다.

본 논문에서는 기존의 문제점들을 해결하기 위해 시간에 따른 객체의 움직임에 대해 방향, 거리, 속도를 고려하여 표현하고 움직임 객체의 궤적에 대한 유사도 검색 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 연구에서 움직임 객체의 궤적 표현과 검색 기법에 대해 기술한다. 3장에서는 제안하는 움직임 객체의 궤적의 표현 방법에 대해 기술하고 4장에서는 궤적 유사도 검색을 위한 연산자와 알고리즘을 기술한다. 마지막으로 5장에서 결론과 향후 연구를 기술한다.

2. 관련연구

[1]에서는 비디오의 논리적 구조 내에 시각적 특징 변화에 의해 객체의 궤적을 표현하였다. [2]에서는 8 가지 방향 관계와 움직임 거리를 이용하여 객체의 궤적(trajjectory)을 표현하고 이를 통해 움직임 방향을 이용한 궤적의 유사도 검색을 제안하였다. 궤적은 시간 간격 $\langle I_1, I_2, \dots, I_m \rangle$ 이 주어져 있을 때, 다음과 같이 표현한다. 이 때, S_i 는 객체의 변위(거리), d_i 는 방향 관계, I_i 는 시간 간격(time interval)을 의미한다.

본 연구는 한국과학재단 목적기초연구(특정기초연구 과제번호 R01-1999-00244)지원으로 수행되었음.

$$\langle (S_1, d_1, I_1), (S_2, d_2, I_2), \dots, (S_m, d_m, I_m) \rangle$$

두 객체 A와 B의 궤적을 $\{ T_1, T_2, \dots, T_m \} (m \geq 1)$, $\{ U_1, U_2, \dots, U_n \} (m \leq n)$ 라 할 때, 궤적 유사도 $TrajSim(A, B)$ 는 (1)과 같다.

$$TrajSim(A, B) = \frac{\max DiffTraj(A, B) - \min DiffTraj(A, B)}{\max DiffTraj(A, B)} \quad (1)$$

$$\min DiffTraj(A, B) = \min \sum_{i=1}^m distance(T_i, U_{i+j}) \quad (\forall j \ 0 \leq j \leq n-i) \quad (2)$$

[3]에서는 각을 이용한 방향 관계와 움직임 거리로 객체의 궤적(trajectory)을 표현하고 방향과 거리를 고려한 궤적의 유사도 검색을 제안하였다. 궤적은 다음과 같이 표현한다. 이 때, α_i 는 방향을 나타내는 각($1^\circ \sim 360^\circ$), D_i 는 시간 간격 I_i 동안 움직인 거리를 1에서 100까지의 상대적 거리로 나타낸 값, I_i 는 시간 간격을 의미한다.

$$\langle (\alpha_1, D_1, I_1), (\alpha_2, D_2, I_2), \dots, (\alpha_m, D_m, I_m) \rangle$$

비디오 데이터베이스에서 단일 움직임 객체 V의 궤적을 $V = \{ V_1, V_2, \dots, V_M \}$, 사용자 질의 Q의 궤적을 $Q = \{ Q_1, Q_2, \dots, Q_N \} (M \geq N)$ 라 할 때, 궤적 유사도 $SDST(V, Q)$ 는 (3)과 같다.

$$SDST(V, Q) = \max \sum_{i=1}^{M-N+1} \left(\frac{\sum_{j=1}^N SR_{i+j}^{(1-\omega_i)} * SD_{i+j}^{(1-\omega_i)}}{N} \right) \quad (3)$$

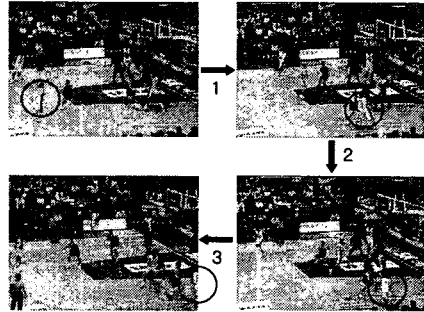
$$SR_i(V_i, Q_i) = \frac{\cos(D_{ang}(V_i, Q_i)) + 1}{2} \quad (4)$$

$$SD_i(V_i, Q_i) = 1 - \frac{|D_{R(i)}^V - D_{R(i)}^Q|}{\max(D_{R(i)}^V - D_{R(i)}^Q)} \quad (5)$$

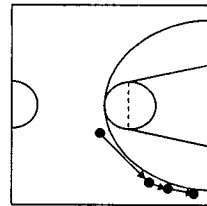
3. 움직임 객체의 궤적 표현

비디오는 수많은 프레임들로 구성된다. 일반적으로 비디오는 1초에 15~45정도의 프레임들로 구성된다. 프레임 내에는 수많은 객체들이 포함되어 있다. 따라서 사용자의 질의에 대해 비디오의 모든 프레임들을 검색하는 것은 어려울 뿐만 아니라 짧은 시간 동안 프레임들에서 객체들의 변화는 미세하기 때문에 모든 프레임에 나타난 객체의 움직임을 표현하는 데는 문제점이 있다. 본 논문에서는 비디오를 구성하는 프레임들 중에서 일정한 시간 간격으로 키프레임(Keyframe)을 추출하고 키프레임에 존재하는 시간에 따른 객체들의 변화를 객체 단위로 나타낸다. [그림 1]은 농구 비디오 데이터를 일정한 시간 간격으로 키

프레임을 추출하여 그 키프레임 내에 존재하는 움직임 객체의 궤적을 표현하기 위해 객체 단위로 표현한 예이다. 궤적의 표현은 경기장을 절대 좌표로 매핑하여 [그림 1]의 (b)와 같이 나타낼 수 있다.



(a) 프레임에서의 객체 움직임 추출



(b) 절대 좌표로 매핑

그림 1. 객체의 궤적 정보 추출 예

움직임 객체가 시간 간격 $\langle I_1, I_2, \dots, I_m \rangle$ 이 주어지고 m 번 움직인다고 할 때, 움직임 각각에 대한 객체의 궤적 정보 $Trajectory_L$ 는 다음과 같이 표현한다. 이 때, R_i, D_i, V_i 는 시간 간격 I_i 동안에 객체가 움직인 방향, 거리, 속도이다. 방향은 $-180^\circ \sim 180^\circ$ 의 각으로 표현한다.

$$Trajectory_L : \langle (R_1, D_1, V_1, I_1), (R_2, D_2, V_2, I_2), \dots, (R_m, D_m, V_m, I_m) \rangle$$

[예제 1]

객체가 시간 간격 $\langle I_1, I_2, I_3, I_4 \rangle$ 동안 움직임 $\langle M_1, M_2, M_3, M_4 \rangle$ 을 가질 때, 궤적 $Trajectory_L$ 은 [그림 2]와 같이 표현할 수 있다.

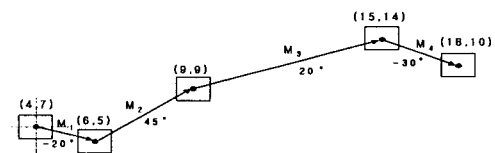


그림 2. 움직임 객체의 궤적 표현

Trajectory_L : <(-20°, 2.8, 0.94, [1,4]), (45°, 5, 1.67, [5,8]), (20°, 7.81, 1.12, [9,16]), (-30°, 5, 1, [17,22])>

4. 궤적 유사도 검색

4.1 검색

객체의 궤적 유사도 검색은 특정 객체와 유사한 궤적을 가지는 객체를 검색한다. 제한하는 궤적 유사도 검색은 객체의 움직임 방향, 거리, 속도를 고려한다. 따라서 궤적 유사도 검색을 위해서 두 객체의 궤적차를 계산하기 위한 두 객체의 움직임 방향차, 거리차, 속도차에 대한 정의가 필요하다. 객체 O_a, O_b 의 존속 구간이 $[t_{sa}, t_{fa}], [t_{sb}, t_{fb}]$ 일 때, 두 객체의 궤적 T_a, T_b 가 아래와 같다고 하자($m \leq n$).

$$\langle (R_a^1, D_a^1, V_a^1, I_a^1), (R_a^2, D_a^2, V_a^2, I_a^2), \dots, (R_a^m, D_a^m, V_a^m, I_a^m) \rangle$$

$$\langle (R_b^1, D_b^1, V_b^1, I_b^1), (R_b^2, D_b^2, V_b^2, I_b^2), \dots, (R_b^n, D_b^n, V_b^n, I_b^n) \rangle$$

[정의 1] 단일 객체의 방향차

객체 O_a 의 궤적이 T_a 일 때, 단일 객체의 방향차 α^i 는 (6)과 같이 정의한다.

$$\alpha^i = \begin{cases} R_{dist_a}^i - 180^\circ, & R_{dist_a}^i > 0 \\ R_{dist_a}^i + 180^\circ, & R_{dist_a}^i < 0 \end{cases} \quad (6)$$

(단, $R_a^0 = 0$ 일 때는 $\alpha^i = R_{dist_a}^i$)

이 때, $R_{dist_a}^i = R_a^i - R_a^{i-1}$ ($1 \leq i \leq m$) 이다.

[예제 2]

단일 객체 O_a 의 방향이 $R_i = \{R_1, R_2, R_3, R_4, R_5\} = \{30^\circ, -45^\circ, 40^\circ, -40^\circ, -140^\circ\}$ 일 때, 움직임 방향차는 아래와 같이 계산하며 [그림 3]과 같이 나타낸다. 나머지도 동일하게 계산한다.

$$\alpha^1 = R_{dist_a}^1 = R_a^1 - R_a^0 = 30^\circ - 0 = 30^\circ$$

$$\alpha^2 = R_{dist_a}^2 + 180^\circ = (R_a^2 - R_a^1) + 180^\circ$$

$$= (-45^\circ - 30^\circ) + 180^\circ = 105^\circ$$

$$\alpha^3 = R_{dist_a}^3 - 180^\circ = (R_a^3 - R_a^2) - 180^\circ$$

$$= (40^\circ - (-45^\circ)) - 180^\circ = -95^\circ$$

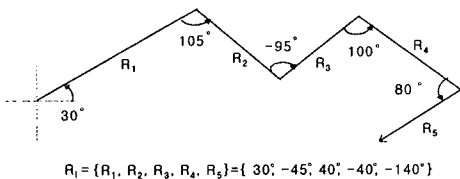


그림 3. 방향차의 예

[정의 2] 두 객체간의 움직임 방향차

두 객체 O_a 와 O_b 의 단일 객체의 방향차를 α^i, β^j 라 할 때, 두 객체간의 움직임 방향차 $RR(\alpha^i, \beta^j)$ 는 (7)과 같이 정의한다.

$$RR(\alpha^i, \beta^j) = \begin{cases} 0.5 + \frac{|\alpha^i - \beta^j|}{360}, & \alpha^i \beta^j < 0 \\ \frac{|\alpha^i - \beta^j|}{360}, & \alpha^i \beta^j > 0 \end{cases} \quad (7)$$

이 때, $|\alpha^i - \beta^j| = \begin{cases} |\alpha^i - \beta^j|, & |\alpha^i - \beta^j| \leq 180 \\ 360 - |\alpha^i - \beta^j|, & |\alpha^i - \beta^j| > 180 \end{cases}$ 이다.

[예제 3]

두 객체 O_a 와 O_b 의 각각 단일 객체의 방향차를 $\alpha^i = \{30^\circ, 20^\circ, -120^\circ\}$ ($i \leq m$), $\beta^j = \{20^\circ, 100^\circ, -120^\circ, -50^\circ\}$ ($j \leq n$)라 할 때, 두 객체의 움직임 방향차는 아래와 같이 계산한다. 이외에 구할 수 있는 방향차들도 동일한 방법으로 계산한다.

$$RR(\alpha^1, \beta^1) = \frac{|\alpha^1 - \beta^1|}{360} = \frac{|20^\circ - 30^\circ|}{360} = 0.027$$

$$RR(\alpha^2, \beta^3) = 0.5 + \frac{|\alpha^2 - \beta^3|}{360}$$

$$= 0.5 + \frac{|20^\circ - (-120^\circ)|}{360} = 0.89$$

[정의 3] 두 객체간의 움직임 거리차

두 객체 O_a 와 O_b 의 움직임 거리를 D_a^i, D_b^j 라 할 때, 두 객체간의 움직임 거리차 $RD(D_a^i, D_b^j)$ 는 (8)과 같이 정의한다.

$$RD(D_a^i, D_b^j) = \frac{|D_a^i - D_b^j|}{MaxD} \quad (8)$$

이 때, $MaxD$ 는 움직임 거리차의 최대값이다.

[정의 4] 두 객체간의 움직임 속도차

두 객체 O_a 와 O_b 의 움직임 속도를 V_a^i, V_b^j 라 할 때, 두 객체간의 움직임 속도차 $RV(V_a^i, V_b^j)$ 는 (9)와 같이 정의한다.

$$RV(V_a^i, V_b^j) = \frac{|V_a^i - V_b^j|}{MaxV} \quad (9)$$

이 때, $MaxV$ 는 움직임 속도차의 최대값이다.

[정의 5] 움직임 궤적차

두 객체 O_a 와 O_b 의 움직임 궤적차 $RDV_{i,j}(O_a, O_b)$ 는 (10)과 같이 정의한다.

$$RDV_{i,j}(O_a, O_b)$$

$$= w_1 RR(\alpha^i, \beta^j) + w_2 RD(D_a^i, D_b^j) + w_3 RV(V_a^i, V_b^j) \quad (10)$$

이 때, w_i 는 가중치로 $w_1 + w_2 + w_3 = 1$ 이 된다. 움직임 궤적차는 0에서 1의 범위를 가지며 차가 적을수록 0에 가깝다.

[정의 6] 객체의 궤적 유사도

두 객체 O_a 와 O_b 의 궤적 유사도 $ST(O_a, O_b)$ 는 (11)과 같이 정의한다.

$$ST(O_a, O_b) = 1 - Tra(O_a, O_b) \quad (11)$$

$$Tra(O_a, O_b) = \min_{\sum_{i=1}^m \frac{RDV_{i,i+k}(O_a, O_b)}{m}} \quad (12)$$

($\forall k, 0 \leq k \leq n - m + 1$)

객체의 궤적 유사도는 0에서 1의 범위를 가지며 유사할 수록 1에 가깝다.

4.2 궤적 유사도 알고리즘

4.1에서 정의한 움직임 객체의 방향, 거리, 속도를 고려한 궤적의 유사도 검색을 위한 알고리즘은 다음과 같다.

```

Trajectory( $T_a, T_b$ )
INPUT :  $T_a = (T_a^1, T_a^2, \dots, T_a^m)$ ,  $T_b = (T_b^1, T_b^2, \dots, T_b^n)$ 
         $\omega_1, \omega_2, \omega_3$ 
OUTPUT : Trajectory_similarity

if(m < n)
    a=m, b=n;
else if(m > n)
    a=n, b=m;

for(int k=0; k <= b-a+1; k++)
{
    for(int i=1; i <= a; i++)
    {
        if(  $\alpha^i \beta^{k+i} < 0$  ) {
            RR =  $0.5 + \frac{|\alpha^i - \beta^{k+i}|}{360}$ ;
        }
        else if(  $\alpha^i \beta^{k+i} > 0$  ) {
            RR =  $\frac{|\alpha^i - \beta^{k+i}|}{360}$ ;
        }
        RD =  $\frac{|D_a^i - D_b^{k+i}|}{MaxD}$ ;
        RV =  $\frac{|V_a^i - V_b^{k+i}|}{MaxV}$ ;
        RVD =  $\omega_1 * RR + \omega_2 * RD + \omega_3 * RV$ ;
        MIN = MIN + RVD;
    }
    MIN = MIN / a;
}
return Trajectory_similarity;
    
```

위의 알고리즘은 사용자 질의 궤적 T_a 와 데이터베이스 내의 단일 움직임 객체의 궤적 T_b 의 유사도를 측정한다. $\omega_1, \omega_2, \omega_3$ 는 방향, 거리, 속도에 대한 가중치이다. 먼저 비교되는 두 궤적의 움직임 수 m 과 n 을 알아내어 두 궤적 중 움직임 수가 작은 것을 기준으로 그 수만큼 비교한다. 비교하는 과정에서 방향차(RR), 거리차(RD) 그리고 속도차(RV)를 구하여 각각에 대해 사용자가 가중치를 주고 계산한 궤적차(RVD)들 중에 가장 작은 값을 궤적차로 한다. 그것을 가지고 두 궤적간의 유사한 정도를 나타내는 궤적 유사도(Trajectory_similarity)를 구한다. 이를 통하여 궤적 유사도 검색을 수행한다.

5. 결론 및 향후 연구

본 논문에서는 비디오 내의 움직임 객체의 궤적을 효과적으로 표현하고 검색할 수 있는 궤적 유사도 검색 기법을 제안하였다. 제안하는 움직임 객체의 궤적 유사도 검색은 객체가 가지는 각각의 움직임에 대해 방향, 거리 그리고 속도를 고려하여 보다 유사한 궤적 검색을 수행한다.

본 논문의 향후 연구 방향으로는 제안한 검색 기법을 실제적인 대용량의 비디오 데이터에 적용하여 성능 평가를 할 예정이다.

[참고문헌]

- [1] Zaher Aghbari, Kunihiko Kaneko and Akifumi Makinouchi, "Modeling and Querying Videos by Content Trajectories," IEEE International Conference on Multimedia and Expo (I) 2000, pp.463-466, 2000.
- [2] J. Z. Li, M.T. Ozsu, and D. Szafron, "Modeling of Moving Objects in a Video Database," Pro. IEEE International Conference on Multimedia Computing and Systems, pp.336-343, 1997.
- [3] Choon-Bo Shim and Jae-Woo Chang, "Spatio-temporal representation and retrieval using moving object's trajectories," ACM Multimedia Workshops 2000, pp.209-212, 2000.
- [4] Zaher Aghbari, Kunihiko Kaneko and Akifumi Makinouchi, "VST-Model: A Uniform Topological Modeling of the Visual-Spatio-Temporal Video Features," ICMCS, Vol. 2, pp.163-168, 1999.