

비디오 데이터베이스에서 이동 객체의 내용 및 궤적 검색

(Content and Trajectory Retrievals of Moving Objects in Video Databases)

북 경 수 [†] 유 재 수 ^{**}

(Kyoung Soo Bok) (Jae Soo Yoo)

요 약 최근 비디오 데이터의 사용이 증가되면서 비디오 데이터에 포함된 이동 객체에 대한 연구가 활발히 진행되고 있다. 비디오 데이터에 포함된 이동 객체는 시간의 변화에 따라 시각적 특징, 공간적인 위치가 변화한다. 또한 이동 객체는 다른 객체 또는 사건과 연관되어 나타난다. 본 논문에서는 비디오 데이터에서 이동 객체에 대한 내용 검색을 지원하기 위한 모델링과 다양한 검색 유형을 제안한다. 제안하는 모델링은 시간의 변화에 따른 객체의 시각적 특징, 이동 궤적 그리고 연관된 의미적 내용을 표현한다. 이를 통해 다양한 검색 유형을 지원한다. 이를 위한 다양한 검색 연산자를 제안한다. 제공하는 모델링의 우수성을 입증하기 위해 검색 시스템을 구현하고 제공되는 질의 유형 관점에서 기존의 기법들과 비교한다. 제안하는 기법은 기존 검색 기법에서 제공하지 않은 다양한 유형의 검색을 지원하며 검색 처리의 효율성을 향상시킨다.

키워드 : 이동 객체, 궤적, 내용 기반 검색, 비디오 데이터베이스

Abstract Recently, together with increasing use of multimedia data, many works on moving objects in video databases have been made. Moving objects change visual features and spatial positions with the lapse of time in video data. And they are related to the other objects or events. In this paper, we propose a new modeling and various query types of moving objects for content based retrieval in video databases. The proposed modeling represents visual features, moving trajectories and semantic contents related to objects. Therefore, it allows to process various query types. And we propose various query operators for the retrieval types. To show the superiority of our modeling, we implement the retrieval systems and compare it with the existing methods in terms of the supporting query types. The proposed method supports various query types and improves the efficiency of the query processing over the existing methods.

Key words : Moving objects, Trajectory, Content Based Retrieval, Video Database

1. 서 론

인터넷의 확산과 컴퓨터 응용 기술의 발달로 인해 멀티미디어 데이터에 대한 사용이 증가되면서 대용량의 멀티미디어 데이터를 효과적으로 저장하고 검색하기 위한 많은 연구들이 진행되고 있다. 특히, 멀티미디어 데

이터의 대표적인 비디오 데이터에 대한 많은 연구들이 진행되고 있으며 많은 검색 시스템들이 개발되었다 [1-3]. 비디오 데이터에는 시청각적 또는 시공간적인 내용은 물론 다양한 유형의 내용들을 서로 연관되어 나타난다. 이러한 비디오 데이터는 대표적인 비정형 데이터로 다양한 유형의 내용들을 검색하기 위해서는 비디오를 효과적으로 표현하기 위한 기법과 이를 효과적으로 검색하기 위한 기법들이 필요하다.

최근 비디오 데이터에 포함된 객체, 사건과 같은 의미적 내용(semantic content)을 표현하고 검색하기 위한 많은 연구들이 진행되고 있다[4,5]. 의미적 내용들은 시간의 변화에 따라 서로 다른 의미적 내용들과 연관 관계를 나타내며 시공간적인 변화를 갖는다. 특히, 객체는

· 이 논문은 2003년도 한국학술진흥재단의 선도연구자 지원사업(KRF-2003-041-D00489)에 의하여 연구되었음

† 학생회원 : 충북대학교 정보통신공학과

ksbok@netdb.chungbuk.ac.kr

** 종신회원 : 충북대학교 전기전자컴퓨터공학부 교수

컴퓨터·정보통신연구소 부장

yjs@cbucc.chungbuk.ac.kr

논문접수 : 2003년 5월 2일

심사완료 : 2004년 3월 2일

특정한 동작을 수행하면서 다른 객체와 함께 서로 다른 의미적 내용에 연관되어 나타난다.

비디오 데이터는 시간의 변화에 따라 공간적인 위치 변화 또는 시각적인 특징 변화를 나타내는 이동 객체(moving object)를 포함하고 있다. 비디오 데이터에 포함된 이동 객체를 검색하기 위해서는 시간의 변화에 따른 객체의 변화를 표현하고 이러한 변화들 사이의 관계성을 기술하기 위한 모델링 기법들이 필요하다. 또한 이러한 이동 객체의 다양한 변화를 효과적으로 검색하기 위한 기법들이 필요하다. [6-8]에서는 비디오 데이터에 나타난 객체들 사이의 시공간 관계(spatio-temporal relation)를 표현하기 위한 관계성을 정의하였다. 이러한 시공간 관계는 시간의 변화에 따른 객체들 사이의 공간적인 관계성을 표현한다. [9-11]에서는 이동 객체의 연속적인 위치 변화를 방향, 거리에 의해 궤적을 표현하고 검색하기 위한 기법들을 제안하였다. 또한 [12]에서는 이동 객체의 궤적을 시각적인 변화를 이용하여 표현하고 검색하는 기법을 제안하였다.

기존의 이동 객체에 대한 검색은 대부분은 객체의 이동 방향만을 고려하거나 시각적 특징 변화의 일부 특성만을 고려하고 있다. 이동 객체는 특정 동작을 수행하면서 서로 다른 의미적 내용에 연관되어 나타난다. 그러나 기존에 제안된 검색 기법은 객체가 시간의 변화에 따라 수행하는 동작 또는 연관된 사건과 같은 의미적 내용을 표현하지 못하기 때문에 이를 통한 의미적 내용 검색을 지원하지 못한다.

본 논문에서는 시간의 변화에 따른 객체의 시각적인 특징과 공간적인 위치 변화뿐만 아니라 이동 객체와 연관된 의미적 내용을 표현하는 새로운 모델링 기법을 제안하고 각각의 내용들을 효과적으로 검색하기 위한 연산자를 제안한다. 또한 제안된 모델링에 표현된 내용에 대한 검색을 수행하기 위해 제안된 연산자들을 이용하여 검색을 수행하는 과정을 제안한다. 이동 객체를 표현하기 위해 먼저 키프레임에서 객체의 시각적인 특징과 공간적인 위치 변화를 표현하고 이를 통해 시간의 변화에 따른 이동 객체의 특성을 표현한다. 이동 객체는 시간의 변화에 따른 객체의 공간적인 이동 변화, 시각적인 특징 변화, 객체에 관련된 의미적 내용에 대한 변화를 표현한다. 제안하는 검색 기법은 새로운 모델링에 기반하여 다양한 내용들의 변화를 표현하기 때문에 이들 내용들 사이의 연관 관계성 및 검색에 필요한 유사도를 계산하기 위한 연산자들이 필요하다. 따라서, 각각의 내용에 대한 유사도를 계산하기 위한 연산자를 제안하고 이를 이용하여 모델링에 표현된 내용들에 대한 검색을 수행하기 위한 기법들을 제안한다. 이동 객체에 대한 공간적인 변화는 객체의 이동 방향, 거리, 속도를 이용하

여 표현하고 유사도를 계산하여 궤적 유사도 검색을 수행하고 시간의 변화에 따른 객체들 사이의 공간적인 위치를 이용하여 거리 유사도 검색을 수행한다. 시각적 특징 궤적에 대한 검색은 프레임에 표현된 칼라, 질감, 모양과 같은 프레임 자체의 시각적 특징에 기반한 유사도 검색과 객체에 대한 시각적 특징 검색을 수행한다. 또한 의미 검색은 비디오 데이터에서 포함된 객체 자체에 대한 검색은 물론 객체가 수행하는 동작과 연관된 사건에 대한 내용을 검색한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 이동 객체에 대한 모델링과 검색 기법에 대해 기술한다. 3장에서는 제안하는 이동 객체 모델링을 기술하고 4장에서는 모델링에 따라 수행하는 다양한 검색 기법을 기술한다. 5장에서는 제안하는 이동 객체 검색을 수행하는 검색 인터페이스에 대해 기술하고 마지막 6장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 관련연구

비디오 데이터에 포함된 이동 객체를 검색하기 위한 기존의 연구들은 크게 이동 객체의 시각적인 특징 변화를 표현하고 검색하기 위한 연구와 공간적인 이동 변화를 표현하고 검색하기 위한 연구로 구분할 수 있다. [9-11]에서는 이동 객체의 공간적인 변화 즉, 이동 객체의 궤적을 이동 방향과 거리에 의해 표현하는 모델링 기법과 궤적을 검색하는 기법을 제안하였다. 이에 반해, [12]에서는 이동 객체의 시각적인 특징 변화를 표현하는 모델링 기법과 유사도 검색을 기법을 제안하였다.

[9]에서는 이동 방향과 거리를 통해 궤적을 표현하고 시간의 변화에 따른 객체들 사이의 시공간 관계성을 정의하였다. [9]에서는 시간 구간 I_i 동안 객체 이동 방향 S_i 과 거리 d_i 을 이용하여 표현한다. 이때, d_i 는 [7]에서 제안한 방향 관계성에 의해 표현한다. [9]에서는 이동 객체의 이동 객체에 대한 궤적 유사도 검색을 수행하기 위해 이동 객체의 방향 관계에 따라 유사도를 정의하고 이를 통해 식 (1)와 같이 궤적 유사도 $TrajSim(A, B)$ 을 계산한다. 이때, $minDiffTraj(A, B)$ 는 이동 궤적의 각 위치들 사이의 차이로 식 (2)와 같이 계산한다. 식 (1)에서 보는 것과 같이 [9]에서는 이동 객체의 방향 관계만을 이용하여 궤적 유사도를 측정하기 때문에 방향 관계에 의해 표현되지 못하는 이동 변화에 대한 검색을 효과적으로 수행하지 못할 뿐만 아니라 궤적 검색 과정에서 이동 객체의 이동 거리를 고려하지 못하는 문제점이 있다. 또한 단순히 이동 객체의 이동 변화에 기반하여 궤적 유사도를 계산하기 때문에 이동 객체들 사이의 공간의 거리 관계 또는 시각적인 특징 기반의 검색을

수행하지 못하는 문제점이 있다.

$$TrajSim(A, B) = \frac{\maxDiffTraj(A, B) - \minDiffTraj(A, B)}{\maxDiffTraj(A, B)} \quad (1)$$

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

[10,14]에서는 구조적 계층인 씬 내에 포함된 이동 객체와 이동 객체들 사이의 관계를 표현하는 모델링 및 검색 기법을 제안하였다. 이동 객체가 특정 시간 동안 연속적인 위치 변화가 나타날 때 객체의 이동 방향 d_i 와 거리 r_i 을 통해 이동 궤적을 표현한다. 이동 방향을 나타내는 d_i 는 8개의 방향을 통해 표현한다[10]. [14]에서는 [10]에서 제안된 이동 객체의 표현을 이용하여 궤적 유사도 검색 기법을 제안하였다. 두 객체 A, B 의 방향과 거리의 변화가 $(d_a, r_a), (d_b, r_b)$ 라 할 때, 두 객체의 방향-거리 유사도 $\zeta(A, B)$ 는 식 (3)과 같이 계산한다. 이때, $\zeta(r_a, r_b)$ 는 씬에 포함된 객체의 이동 방향의 차를 나타낸다. 객체들 사이의 방향 차를 나타내는 $\zeta(r_a, r_b)$ 는 0~180의 값으로 식 (4)와 같이 계산된다. 이때, $MaxDir$ 는 두 개의 최대 방향 차를 나타내는 180이다.

$$\zeta(A, B) = \min(d_a, d_b) \times \zeta(r_a, r_b) \quad (3)$$

$$\zeta(r_a, r_b) = 100 \times \frac{MaxDir - D(r_a, r_b)}{MaxDir} \quad (4)$$

[14]에서는 두 객체의 방향-거리 유사도를 통해 두 객체의 궤적 T_1 과 T_2 가 주어져 있을 때 객체 사이의 궤적 유사도를 정의한다. 먼저 두 궤적의 시공간적으로 가능한 모든 경우를 고려하여, T_1 이 T_2 를 우선하면서 겹치는 경우에는 식 (5)와 같이 S_1 을 계산하고 T_1 가 T_2 내에 포함되는 경우에는 식 (6)과 같이 S_2 을 계산한다. 또한 T_2 가 T_1 을 우선하면서 겹치는 경우에는 식 (7)과 같이 S_3 을 계산한다.

$$S_1 = \max \left(\frac{\sum_{(i,j)=(1,m-n+k)}^{(n+1-k,m)} C((d_{pi}, r_{pi}), (d_{ij}, r_{ij}))}{\sum_{i=1}^n d_{pi}}, \forall k=1, \dots, n \right) \quad (5)$$

$$S_2 = \max \left(\frac{\sum_{(i,j)=(k,k)}^{(n+k,n+k)} C((d_{pi}, r_{pi}), (d_{ij}, r_{ij}))}{\sum_{i=1}^n d_{pi}}, \forall k=1, \dots, n \right) \quad (6)$$

$$S_3 = \max \left(\frac{\sum_{(i,j)=(k+1,1)}^{(n,n-k)} C((d_{pi}, r_{pi}), (d_{ij}, r_{ij}))}{\sum_{i=1}^n d_{pi}}, \forall k=1, \dots, n \right) \quad (7)$$

두 객체의 궤적 T_1 과 T_2 의 궤적 유사도는 식 (8)과 같이 정의한다. 만약 두 궤적 T_1 과 T_2 가 하나의 선분과 같다면 궤적 유사도는 $\zeta(r_a, r_b)$ 과 같다.

$$\zeta(T_1, T_2) = \max(S_1, S_2, S_3) \quad (8)$$

그러나 [14]에서 제안된 기법은 이동 객체의 변화가 비디오의 논리적 계층 구조인 씬 내에 의해 표현되기 때문에 객체의 연속적인 변화를 표현하지 못하는 문제점이 있다. 또한 이동 객체의 공간적인 위치 관계와 시각적인 특징을 표현하지 않기 때문에 다양한 유형의 이동 객체 변화를 검색하지 못하는 문제점이 있다.

[11]에서는 시공간 관계성과 더불어 특정 시간 구간 동안 움직인 객체의 이동 거리를 고려하여 이동 객체의 궤적을 표현하는 모델링 기법을 제안하였다. 이동 방향 a_i 은 객체의 중심들 사이의 0~360 사이의 각을 이용하여 표현한다. 또한 이동 거리 D_i 는 객체의 상대적인 이동 거리로 0~100 사이로 표현된다. 사용자의 질의에 대한 이동 객체의 궤적 유사도를 계산하기 위해 질의에 대한 객체의 방향 차 $SR(VS_i, SQ_i)$ 와 거리 차 $SD(VS_i, SQ_i)$ 를 식 (9)와 식 (10)과 같이 계산한다. 질의에 대한 객체의 궤적 유사도는 식 (11)과 같이 식 (9)와 식 (10)에서 계산된 값을 유사도를 정의한다. 이때, w_1, w_2 는 방향 차와 거리 차에 부여된 가중치를 나타낸다. 이러한 기법은 객체의 이동 방향과 이동 거리를 고려하기 때문에 기존의 기법에 비해 보다 정확한 유사도를 계산할 수 있으나 객체의 공간적인 거리 관계와 객체의 시각적인 특성 변화를 검색하지 못하는 문제점이 있다.

$$SR(VS_i, SQ_i) = \frac{\cos(D_{avg}(VS_i, SQ_i)) + 1}{2} \quad (9)$$

$$SD(VS_i, SQ_i) = 1 - \frac{|VS_i - SQ_i|}{\max(VS_i, SQ_i)} \quad (10)$$

$$SDST(VS, QS) = \max \left\{ \frac{\sum_{i=1}^N SR_{i+j}^{(1-w_1)} * SD_{i+j}^{(1-w_2)}}{N} \right\} \quad (11)$$

[12]에서는 비디오를 OL (Object Level), FL (Frame Level) 그리고 SL (Shot Level)의 세 개의 논리적인 계층 구조로 나누고 샷 계층에 나타나는 객체의 시각적 특징 변화를 이동 객체의 궤적으로 표현하였다. OL 계층은 하나의 프레임에 나타나는 객체들에 대한 색상, 모션(motion) 그리고 위치 정보를 표현한다. 객체의 색상 정보는 칼라 히스토그램에 의해 표현하고 객체가 움직이는 방향과 속도를 나타낸다. 또한 객체의 위치 정보는 프레임에서 객체의 절대적인 위치를 MBR 로 표현한다. FL 계층은 프레임에 포함된 객체들 사이의 관계를 VST 관계성에 의해 표현한다[13]. ST 계층은 프레임

에 포함된 객체들에 대한 특징 궤적 F , 관계 궤적 R 그리고 내용 궤적 θ 를 표현한다. F 는 샷에 존재하는 각 프레임에 나타난 객체의 색상, 동작, 위치 정보에 대한 리스트로 나타낸다. [12]에서는 시간의 변화에 따른 이동 객체의 시각적 특징 변화에 따른 유사도 검색을 수행한다. 샷 내에 포함된 객체 O_i 와 O_j 의 색상 히스토그램을 I_i , J_i 라 하고 I_i 와 J_j 의 유사도를 a_{ij} 라 할 때, 유사도 검색을 수행하기 위해서 샷 내에 포함된 객체들에 대한 색상과 동작에 대한 차 $color_dist(I_i, J_i)$, $motion_dist(I_i, J_i)$ 은 식 (12)와 같이 계산하고 계산된 거리 차의 평균을 계산하여 avC 와 avM 에 기록한다. 또한 [13]에서 정의한 VST 관계에 대한 유사도를 이용하여 VST_rel_dist 를 계산하여 그 평균을 avR 에 기록한다. 이렇게 계산된 avC , avM , avR 에 가중치를 부여하여 식 (13)과 같이 실제적인 유사도를 계산한다. [12]에서 제안된 기법은 비디오의 논리적 계층 구조에 의해 변화되는 이동 객체의 특성을 고려하기 때문에 연속적인 객체의 변화를 효과적으로 검색하지 못할 뿐만 아니라 객체의 공간적인 특성 변화를 검색하지 못하는 문제점이 있다.

$$d = \left| \sum_{i=1}^m [(I_i - J_i) + \sum_{j=1}^n a_{ij} * (I_j - J_j)] \right| \quad (12)$$

$$packet_dist = (avC * weight1) + (avM * weight2) + (avR * weight3) \quad (13)$$

따라서 본 논문에서는 기존에 제안된 이동 객체의 검색의 문제점을 해결하기 위해 다음과 같은 특성을 고려하여 연속적인 이동 객체의 변화를 검색하기 위한 기법을 제안한다. 제안하는 검색 기법은 연속적인 객체의 변화를 효과적으로 검색하기 위해 비디오의 구조적 정보와 관계없이 이동 객체의 변화를 표현하고 객체의 공간적인 위치뿐만 아니라 위치 변화를 표현하고 검색한다. 또한 시간에 따른 시각적 특징 변화를 표현하여 특정 시점 또는 객체 전체의 시각적 유사도를 검색을 지원한다. 이동 객체의 공간적인 변화에 기반하여 공간적인 위치 또는 위치 변화를 이용하여 거리 유사도 검색을 수행하고 이동 변화의 궤적을 검색하기 위해 객체의 이동 방향, 거리, 속도를 고려한 궤적의 유사도 검색 수행할 수 있도록 한다.

3. 이동 객체 모델링

비디오는 수많은 프레임들로 구성되어 있다. 보통 디지털 비디오는 1초에 15~45 프레임들이 포함되어 있다. 하나의 객체를 표현하기 위해 각각의 프레임에 나타난 객체의 변화는 매우 미세하기 때문에 사용자에 의해 인

식되지 못할 뿐만 아니라 그 변화를 모두 표현하는 것은 시간적으로 문제가 있다. 따라서 제안하는 모델링에서는 모든 프레임에 나타난 객체들의 이동 변화를 통해 객체의 궤적을 표현하는 것이 아니라 특정 시간 단위로 키프레임을 추출하고 이를 통해 이동 객체를 표현한다. 그림 1은 비디오 데이터에서 이동 객체를 모델링하는 과정을 나타낸 것이다. 이동 객체를 모델링하기 위해서는 먼저 원시 데이터에서 특정 시간 t 간격으로 키프레임을 추출하여 이동 객체의 시각적인 특징과 공간적인 위치를 추출한다. 이를 통해 이동 객체의 시각적 특징 변화와 공간적인 위치 변화를 표현한다. 또한 이동 객체의 의미적 내용을 추출하기 위해서는 의미적 지식 데이터베이스(Semantic Knowledge Database)에 기록된 의미적 내용을 이용하여 추출된 객체에 대한 의미적 내용을 표현한다.

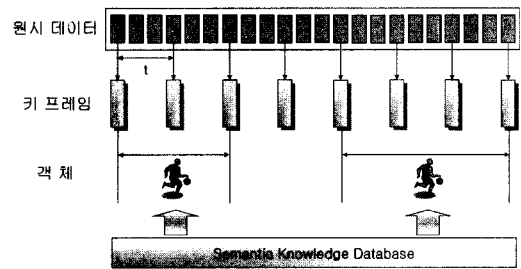


그림 1 모델링 과정

이동 객체를 표현하기 위해 먼저 특정 시간 단위로 키프레임을 추출하고 추출된 키프레임에 포함된 객체의 특성을 표현한다. 이러한 키프레임 KF 에는 식 (14)와 같이 키프레임 자체에 대한 특징과 키프레임에 나타난 객체들에 대한 정보를 나타낸다. 이때, F 는 키프레임 식별자, OL 은 프레임 내에 나타난 객체에 대한 정보 리스트 그리고 VF 는 해당 키프레임의 시각적 특징 정보를 나타낸다.

$$KF : \langle F, OL, VF \rangle \quad (14)$$

키프레임에 나타난 객체에 대한 정보를 나타내는 OL 은 식 (15)와 같이 키프레임에 나타난 객체에 대한 시각적인 특징과 공간적인 특징을 표현한다. OL_i 는 프레임에 나타난 하나의 객체에 대한 정보를 나타낸 것으로 식 (16)과 같이 객체 이름 ON_i , 객체의 공간적 위치 MBR_i , 객체의 중심점 C_i , 객체의 윤곽 정보를 S_i 그리고 객체의 시각적 특징 정보를 RGB 값으로 나타낸 OF_i 로 의해 표현한다.

$$OL : \langle OL_1, OL_2, \dots, OL_n \rangle \quad (15)$$

$$OL_i : \langle ON_i, MBR_i, C_i, S_i, OF_i \rangle, 1 \leq i \leq n \quad (16)$$

이동 객체는 키프레임에 대한 시각적인 특징, 공간적인 위치 정보 그리고 객체에 연관된 의미적 내용을 기반으로 연속된 시간의 변화에 따른 객체의 변화를 나타낸다. 시간의 변화에 따른 이동 객체의 연속적인 변화는 시각적인 특징 변화에 대한 궤적, 공간적인 위치 변화를 나타내는 이동 궤적 그리고 객체가 수행한 동작과 이를 통해 연관된 사건에 대한 의미적 내용에 대한 리스트에 의해 표현한다. 이동 객체의 표현은 식 (17)과 같다. ON 는 객체의 이름, O_i 는 객체에 부여된 고유한 식별자로 동일한 이름을 갖는 객체가 연속된 시간 구간에 나타나지 않을 경우 서로 다른 식별자를 부여한다. CL , ML 그리고 TL 은 시간의 변화에 따른 객체의 시각적 특징 변화, MBR 변화, 이동 궤적을 나타낸다. 또한 AL 과 EL 은 객체가 수행한 동작 리스트와 연관된 사건의 리스트를 나타낸다. I_n 는 객체가 나타난 존속 구간 $[t_s, t_e]$ 로 객체의 등장 시간 t_s 와 소멸 시간 t_e 에 의해 표현한다. CL , ML 그리고 TL 은 독립적으로 표현될 수 있으나 AL 과 EL 은 항상 연관되어 표현한다.

$$O : \langle ON, O_i, CL, ML, TL, AL, EL, I \rangle \quad (17)$$

객체의 시각적 특징 변화를 나타내는 CL 는 식 (18)과 같이 I_i 동안 객체의 색상 정보를 나타내는 RGB 값을 나타내는 CL_i 의 리스트이다. 또한 공간적인 위치와 크기 변화를 나타내는 ML 는 식 (19)과 같이 표현한다. 이때, I_i 동안에 객체의 MBR 을 나타내는 MBR_i , 객체의 중심점 C_i 그리고 객체의 MBR_i 에 포함된 객체의 윤곽 정보 S_i 로 표현한다.

$$CL : \langle (CL_1, I_1), (CL_2, I_2), \dots, (CL_n, I_n) \rangle \quad (18)$$

$$ML : \langle (MBR_1, C_1, S_1, I_1), (MBR_2, C_2, S_2, I_2), \dots, (MBR_n, C_n, S_n, I_n) \rangle \quad (19)$$

객체는 연속된 시간의 변화에 따라 다양한 동작을 수행하면서 여러 사건에 연관되어 비디오에 나타난다. AL 은 객체가 연속된 시간 동안 수행한 동작 리스트로 식 (20)과 같이 표현한다. 이 때, ACT_i 는 I_i 동안에 객체가 수행한 동작이다. EL 은 객체가 연관된 사건의 리스트로 식 (21)과 같이 표현한다. 이 때, EVT_i 는 I_i 동안에 객체가 연관된 사건을 나타낸다.

$$AL : \langle (ACT_1, I_1), (ACT_2, I_2), \dots, (ACT_n, I_n) \rangle \quad (20)$$

$$EL : \langle (EVT_1, I_1), (EVT_2, I_2), \dots, (EVT_n, I_n) \rangle \quad (21)$$

이동 객체는 시간에 따라 공간상에서 위치나 모양이 변하는 객체이다. 이동 객체의 시공간적 특성을 표현하기 위해서 시간에 따른 객체의 이동에 대한 정보를 나타내는 궤적 정보의 표현이 필요하다. 시간 간격 $\langle I_1, I_2, \dots, I_m \rangle$ 이 주어질 때 이동 객체의 이동 궤적

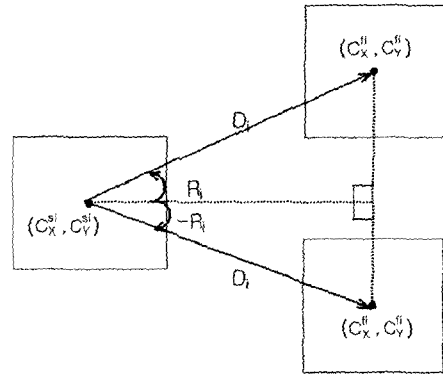


그림 2 객체의 이동 방향

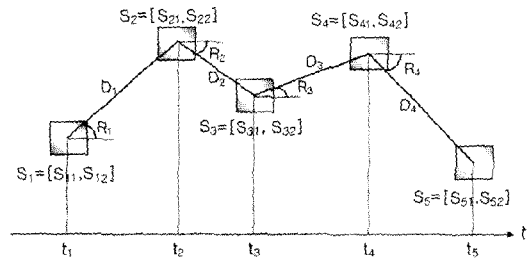


그림 3 이동 객체의 궤적

TL 는 식 (22)와 같이 I_i 동안 객체가 이동 방향, 거리, 속도를 나타내는 R_i, D_i, V_i 에 의해 나타낸다. 이동 방향을 나타내는 R_i 는 그림 2와 같이 객체가 중심점 사이의 이동 방향을 $-180 \sim 180$ 범위의 각으로 나타낸다. D_i 는 이전 위치의 중심에서 다음 위치의 중심까지의 이동 거리에 의해 표현되며 V_i 는 I_i 동안 객체가 이동한 거리 D_i 의 평균 속도를 나타낸다.

$$TL : \langle (R_1, D_1, V_1, I_1), (R_2, D_2, V_2, I_2), \dots, (R_m, D_m, V_m, I_m) \rangle \quad (22)$$

예를 들어, 그림 3과 같이 네 번의 움직임 변화를 갖는 이동 객체 O 가 있다고 가정하자. 각각의 이동 객체의 중심이 $S_i = [S_{i1}, S_{i2}]$ 이고 R_i 를 사이 각이라 할 때, 이동 객체의 궤적은 아래와 같다. 이때, 이동 객체의 속도를 나타내는 $V_i = \frac{D_i}{t_{i+1} - t_i}$ 이다.

$$\langle (R_1, D_1, V_1, [t_1, t_2]), (-R_2, D_2, V_2, [t_2, t_3]), (R_3, D_3, V_3, [t_3, t_4]), (-R_4, D_4, V_4, [t_4, t_5]) \rangle$$

4. 제안하는 이동 객체 검색

본 장에서는 이동 객체의 검색을 수행하기 위한 유사도를 계산하기 위한 연산자와 연산자를 통해 논문에서

지원하는 검색 유형을 수행하는 과정을 기술한다. 본 논문에서 제안하는 검색은 내용 검색과 궤적 검색을 구분한다. 내용 검색은 이동 객체에 연관된 의미적 내용에 기반한 의미적 내용 검색과 시각적인 특징 검색을 수행한다. 궤적 검색은 시간의 변화에 따른 객체의 공간적인 위치에 기반하여 이동 객체들 사이의 거리 유사도 검색과 연속적인 이동 변화를 이용한 이동 방향, 거리 속도에 기반한 궤적 유사도 검색을 수행한다. 이러한 검색들은 독립적인 검색으로 수행할 수도 있지만 각각의 검색 유형에 가중치를 부여하여 가중치 혼합검색을 수행한다. 본 논문에서 제공하는 검색 유형은 아래와 같이 의미적 내용 검색, 시각적 특징 기반의 검색, 거리 유사도 검색, 궤적 유사도 검색을 지원한다.

- 검색 유형 1 : “주어진 동작을 수행하는 객체를 검색하라” 또는 “주어진 사건에 연관된 객체를 검색하라” 등과 같은 이동 객체에 대한 의미적 내용 검색
- 검색 유형 2 : “주어진 프레임과 유사한 프레임에 포함된 객체를 검색하라” 또는 “주어진 색상과 유사한 색상을 갖는 객체를 검색하라” 등과 같이 시각적 특징 기반의 검색
- 검색 유형 3 : “주어진 객체와 근접한 거리에 존재하는 객체를 포함하는 프레임을 검색하라” 또는 “주어진 객체와 근접한 거리에 존재하는 객체를 검색하라” 등과 같은 이동 객체들 사이의 거리 유사도 검색
- 검색 유형 4 : “주어진 객체와 유사한 궤적을 포함하는 객체를 검색하라”와 같은 이동 객체에 대한 궤적 유사도 검색

4.1 의미적 내용 검색

의미적 내용 검색은 비디오 데이터에서 포함된 객체 자체에 대한 검색은 물론 객체가 수행하는 동작과 연관된 사건에 대한 내용을 검색한다. 하나의 객체는 객체의 이름, 객체 식별자 그리고 존속 구간에 의해 표현하였다. 또한 객체가 시간의 변화에 따라 수행한 동작과 사건에 대한 내용을 표현하였다. 이를 통해 특정 객체가 수행한 동작 또는 특정 사건에 연관된 객체를 검색할 수 있다. 일반적으로 의미 검색은 의미적 내용 자체에 대한 검색 뿐만 아니라 시간 관계성에 기반한 검색을 수행할 수 있다. 의미 검색에서 제공되는 검색 형태는 다음과 같다.

- 객체 *O* 이전에 또는 이후에 존재하는 객체를 검색하라.
- 사건 *E*와 연관된 객체를 검색하라.
- 객체 *O* 이전에 존재하는 객체 중 사건 *E*에 연관된 객체를 검색하라.
- 객체 *O* 이후에 존재하는 객체 중 동작 *A*를 수행하는 객체를 검색하라.

그림 4는 의미 검색을 수행하는 semantic_search()

```

Algorithm semantic_search(E, A, R)
Input : E = 검색할 사건, A = 객체에 대한 동작, R = 시간 관계
Output : O = 검색된 객체 리스트
{
    i=0; k=0; m=0;
    if(E가 존재한다면){
        사건 검색을 위한 검색 조건을 확인;
        while(검색할 객체가 존재할 때){
            result[i]=object_event_search(); //객체와 연관된 사건을 검색
            i++;
        }
        if(A가 존재한다면){
            result[]에 저장된 내용을 compare[]에 저장;
            동작 검색을 위한 질의를 확인;
            for(j=0; j=i; j++){ //compare[]에 존재하는 객체에 대해 동작에 대한 검색
                result[k]=object_action_search(); //사건과 대한 객체에 대한 동작을 검색
                k++;
            }
        }
    }
    if(R이 존재한다면){
        result[]에 저장된 내용을 compare[]에 저장;
        for(j=0; j=k; j++){
            compare[]에 존재하는 객체에 대한 존속 구간을 판별
            result[m]=temp_search(); //객체의 존속 구간과 R에 대한 조건을 비교
            m++;
        }
    }
    result[]에 존재하는 결과 값을 반환;
}
    
```

그림 4 의미 검색

알고리즘을 나타낸 것이다. 의미 검색을 수행하기 위해서 제시된 검색 조건을 판별하기 위해 입력된 내용이 무엇인지를 판별한다. 사용자에게 의해 입력된 검색 조건이 판별되면 주어진 검색을 수행하기 위해 object_event_search()과 object_action_search()을 통해 사건 또는 객체가 수행한 동작에 대한 검색을 수행한다. 만약 사용자가 사건과 함께 객체에 대한 동작을 함께 검색하고자 할 때는 object_event_search()을 통해 검색된 결과 내에 포함된 객체에 대해 다시 사건에 대한 검색을 수행한다. 또한 시간 관계성에 대한 검색을 요청하는 경우에는 사건 또는 동작에 검색 결과에 대해 temp_search()를 사용한다.

4.2 시각적 특징 검색

시각적 특징 검색은 프레임 자체에 대한 시각적 특징은 물론 객체의 시각적 특징에 기반한 검색을 수행한다. 프레임 자체에 대한 시각적 특징 기반의 검색은 프레임에 표현된 칼라, 질감,모양과 같은 프레임 자체의 시각적 특징에 기반한 유사도 검색을 수행한다. 또한 객체에 대한 시각적 특징 검색은 시간의 변화에 따른 객체의 시각적 변화를 나타내는 RGB 값을 통해 유사도 검색을 수행한다. 시각적 특징 기반의 검색에 제공되는 검색 유형은 다음과 같다.

- 주어진 질의 이미지 *Q*와 유사한 프레임 *F*을 검색하라.
- 주어진 색상 *Q*와 유사한 색상을 갖는 객체 *O*을 검색하라.
- 객체 *O*를 포함하는 프레임에서 주어진 질의 이미지 *Q*와 유사한 프레임을 검색하라.

• 주어진 질의 이미지 Q 에 포함된 객체 O 의 시각적 특징과 유사한 객체를 검색하라.

4.2.1 프레임의 시각적 특징 검색

비디오에서 추출한 키프레임에는 프레임 자체의 시각적 특징을 표현하였다. 이를 기반으로 프레임에 대한 시각적 특징 기반 유사도 검색을 수행한다. 프레임에 대한 시각적 특징 기반의 유사도 검색을 수행하기 위해서는 먼저 주어진 질의 이미지 Q 와 프레임 F 의 시각적 특징에 대한 유사도 $SF(Q, F)$ 을 정의한다. 주어진 질의 이미지 Q 와 프레임 F 의 칼라, 질감, 모양에 대한 시각적 특징이 $\langle C_Q, T_Q, S_Q \rangle$ 와 $\langle C_F, T_F, S_F \rangle$ 일 때, 프레임의 시각적 특징 유사도 $SF(Q, F)$ 는 식 (23)과 같다. 시각적 특징에 대한 유사도 $SF(Q, F)$ 는 0~1 사이의 범위를 갖으며 질의 이미지와 프레임의 시각적 특징이 유사할수록 1에 가깝다. $SF(Q, F)$ 에서는 시각적 특징 기반의 유사도를 0~1 사이의 범위를 갖도록 하기 위해 질의와 프레임의 시각적 특징 차 $D_F(Q, F)$ 를 프레임의 최대 시각적 특징 차를 나타내는 $MaxF$ 로 나누어 계산한다.

$$SF(Q, F) = 1 - \frac{D_F(Q, F)}{MaxF} \quad (23)$$

$D_F(Q, F)$ 는 질의 이미지 Q 와 프레임 F 의 시각적 특징 차이로 $MaxF$ 는 프레임의 시각적 특징의 최대차이다. 식 (24)는 $D_F(Q, F)$ 을 계산한 것으로 질의 이미지의 시각적 특징과 프레임의 시각적 특징의 차를 계산한 것이다. 또한 ω_α 는 시각적 특징에 부여된 가중치로 $\omega_\alpha + \omega_\beta + \omega_\gamma = 1$ 이다.

$$D_F(Q, F) = \sqrt{\omega_\alpha(C_Q - C_F)^2 + \omega_\beta(T_Q - T_F)^2 + \omega_\gamma(S_Q - S_F)^2} \quad (24)$$

그림 5는 프레임에 대한 시각적 특징 검색을 수행하는 `frame_search()` 알고리즘을 나타낸다. `frame_search()`는 질의 이미지에 대한 시각적 특징을 추출하기 위해 `calculate_image()`를 수행한다. `image_search()`는 식 (23)을 이용하여 질의 이미지와 각각의 키프레임의 시각적 특징을 비교한다.

```

Algorithm frame_search(Q, W, R)
Input : Q = 질의 이미지, W = 가중치, R = 검색 유사도
Output : F = 검색된 프레임의 리스트
{
    i=0;
    calculate_image(); //질의 이미지에 대한 시각적 특징을 계산
    while(검색할 프레임이 존재할 때){
        sim=image_search(); //질의 이미지와 프레임의 시각적 특징을 비교
        if(sim>=R){
            result[i]에 비교된 프레임을 저장;
            i++;
        }
    }
    result[]에 저장된 결과를 반환;
}
    
```

그림 5 프레임의 시각적 특징 검색

적 특징을 비교한다.

4.2.2 객체의 시각적 특징 검색

객체 계층은 키프레임 내에 포함된 객체들에 대해 시간의 변화를 이용하여 시각적인 특징 궤적을 표현하였다. 이를 통해 객체의 시각적 특징 기반의 유사도 검색을 수행한다. 사용자가 선택한 색상 Q 에 대한 객체의 유사도 검색 $SC(Q, O)$ 을 정의한다. 객체 O 에 대한 시각적 특징을 나타내는 CL 이 $\langle (CL_1, I_1), (CL_2, I_2), \dots, (CL_n, I_n) \rangle$ 이고 주어진 색상 Q 의 시각적 특징이 $\langle R_Q, G_Q, B_Q \rangle$ 라 할 때, $SC(Q, O)$ 는 식 (25)와 같다. $SC(Q, O)$ 는 사용자의 질의 색상에 대해 가장 유사한 색상을 많이 포함하고 있는 객체를 검색한다. $SC(Q, O)$ 는 0~1 사이의 범위를 갖으며 질의 색상과 유사한 색상을 많이 포함하고 있을수록 1에 가깝다.

$D_c(Q, O)$ 는 질의 색상 Q 와 객체의 시각적 특징 변화에 대해 계산한 것으로 식 (26)과 같이 계산한다. 객체 O 의 존속 구간이 $[t_s, t_f]$ 이고 객체의 시각적 특징 리스트 CL 에서 i 번째 리스트의 존속 구간이 $[t_s, t_f]$ 라 할 때, $D_c(Q, O)$ 는 주어진 질의 Q 에 대해 시각 리스트에서 가장 유사한 색상을 포함하고 있는 i 번째 리스트와의 거리 차를 계산한다. 이때, $MaxC$ 는 객체와 질의 색상의 시각적 특징의 최대 차이이다. 또한 ω_α 는 RGB 값에 부여된 가중치로 $\omega_\alpha + \omega_\beta + \omega_\gamma = 1$ 이다.

$$SC(Q, O) = 1 - \frac{D_c(Q, O)}{MaxC} \quad (25)$$

$$D_c(Q, O) = \text{Min} \left(1 - \frac{t_f - t_s}{t_j - t_i} \right) \sqrt{\omega_\alpha(R_Q - C_i) + \omega_\beta(G_Q - Q_i) + \omega_\gamma(B_Q - B_i)}, \quad (26)$$

$$1 \leq i \leq n$$

그림 6은 객체에 대한 시각적 특징 검색을 위한 `color_search()` 알고리즘을 나타낸 것이다. 시각적 특징

```

Algorithm color_search(Q, W, R)
Input : Q = 질의 색상, W = 가중치, R = 검색 유사도
Output : O = 검색된 객체의 리스트
{
    i=0;
    while(검색할 객체가 존재할 때){
        object_interval(); // 객체의 존속 구간을 계산
        while(객체의 존속 구간 동안){
            calculate_object_color(); // 객체의 시각적 특징 변화를 계산
            sim=object_color(); // 질의 색상과 객체에 대한 비교
            if(sim>=R){
                result[i]에 비교된 객체를 저장;
                i++;
            }
        }
    }
    result[]에 저장된 결과를 반환;
}
    
```

그림 6 객체의 시각적 특징 검색

검색을 수행하기 위해서는 먼저 질의 질의 선택된 색상이 판별하고 객체의 존속 구간 내에서 색상 변화를 calculate_object_color()로 계산한다. 질의 및 객체의 색상 변화를 판별하면 객체의 색상 유사도 검색을 수행하기 위해 object_color()를 수행한다. object_color()는 식 (25)를 이용하여 유사도를 계산한다.

4.3 거리 유사도 검색

거리 유사도 검색은 시간의 변화에 따라 객체들 사이의 공간적인 위치가 근접한 객체를 검색한다. 즉, 객체 계층에 표현된 MBR 리스트를 이용하여 시간에 따른 공간적인 MBR에 대한 거리를 계산하여 질의 MBR에 근접한 객체를 검색한다. 거리 유사도 검색에서 제공되는 질의 유형은 다음과 같다.

- 주어진 질의 Q에 근접한 객체 O를 검색하라.
- 주어진 객체 Q와 근접한 위치에서 움직이는 객체 O를 검색하라.
- 주어진 질의 Q와 가장 근접한 거리에 존재하는 객체 O를 포함하는 프레임을 검색하라.

공간적인 거리 유사도를 계산하기 위해 먼저 특정 시점에서 객체들의 MBR 사이의 거리를 정의하고 질의에 대한 객체의 거리 유사도를 정의한다. 특정 시점 t에서 질의 Q에 대한 객체의 거리를 정의하기 위해 질의 Q와 객체 O의 MBR이 MBR'_Q=(X'_Q, Y'_Q)과 MBR'_O=(X'_O, Y'_O)라 하자. 이때, X'_Q=(X'_{SQ}, X'_{FQ}), Y'_Q=(Y'_{SQ}, Y'_{FQ}), X'_O=(X'_{SO}, X'_{FO}), Y'_O=(Y'_{SO}, Y'_{FO})이며 X'_{SQ}와 X'_{FQ}는 질의 Q에 대한 X축의 시작 위치와 끝 위치, X'_{SO}와 X'_{FO}는 객체 O에 대한 X축의 시작 위치와 끝 위치를 나타낸다. 이와 유사하게 Y'_{SQ}, Y'_{FQ}, Y'_{SO}, Y'_{FO}는 질의 Q와 객체 O에 대한 Y축의 시작 위치와 끝 위치를 나타낸다. 특정 시점 t에서 MBR들 사이의 D_{MBR}(Q, O, t)은 수식 (27)과 같이 정의한다.

$$D_{MBR}(Q, O, t) = \begin{cases} \sqrt{X^2 + Y^2} & , t \in [t_s^Q, t_f^Q], [t_s^O, t_f^O] \\ MaxDist & , else \end{cases} \quad (27)$$

이 때,

$$X = \begin{cases} X'_{SO} - X'_{FQ} & , X'_{SO} > X'_{FQ} \\ 0 & , X'_{SO} < X'_{FQ} \\ else & , else \end{cases} , Y = \begin{cases} Y'_{SO} - Y'_{FQ} & , Y'_{SO} > Y'_{FQ} \\ 0 & , Y'_{SO} < Y'_{FQ} \\ else & , else \end{cases}$$

이다. MaxDist는 객체와 질의 사이의 최대 거리를 나타내며 질의에 대해 객체가 존속 구간 내에 포함되어 있지 않을 경우 MaxDist를 나타낸다.

그림 7은 객체와 질의에 대한 MBR 사이의 거리를 나타낸 예이다. 객체 O에 대한 질의 Q_1, Q_2, Q_3, Q_4에 대한 MBR 사이의 거리는 질의와 객체의 MBR의 영역 경계면 사이의 최소 거리로 정의한다. 만약 질의 Q_1과 같이 객체 O와 공간적인 겹침이 발생한다면 D_{MBR}(Q_1, O, t)

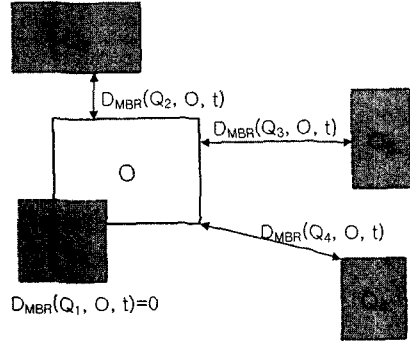


그림 7 특정 시점에서 MBR들 사이의 거리

의 거리는 0이 된다.

특정 시점 t에서 질의 Q와 객체 O의 MBR이 MBR'_Q=(X'_Q, Y'_Q)과 MBR'_O=(X'_O, Y'_O)일 때, 특정 시점 t에서 거리 유사도 SD(Q, O, t)는 식 (28)과 같이 정의한다. SD(Q, O, t)는 0~1 사이의 범위를 가지며 질의 Q에 근접하게 존재할수록 1에 가깝다.

$$SD(Q, O, t) = 1 - \frac{D_{MBR}(Q, O, t)}{MaxDist} \quad (28)$$

이제 질의 Q에 대한 객체 O의 거리 유사도 SD(Q, O)을 정의하자. 질의 Q와 객체 O의 존속구간이 [t_s^Q, t_f^Q], [t_s^O, t_f^O]라 할 때, 질의에 대한 객체의 거리 유사도 SD(Q, O)는 식 (29)와 같이 정의한다. SD(Q, O)는 질의에 대한 이동 객체의 MBR 사이의 거리를 비교하기 위해 두 객체가 가지는 존속 구간 중에 최대인 객체의 소멸 시간 t_f와 최소인 객체의 등장시간 t_s의 시간 구간으로 나누어 계산한다. SD(Q, O)는 0~1 사이의 범위를 가지며 질의에 대한 객체의 공간적인 위치가 근접할수록 1에 가깝다. 만약 한 객체가 존속하는 구간에 다른 객체가 존속하지 않아 두 객체의 MBR 거리를 계산할 수 없을 때는 최대 거리 값인 MaxDist값을 가진다.

$$SD(Q, O) = \frac{\sum_{t=t_s}^{t_f} D_{MBR}(Q, O, t)}{t_f - t_s}, \quad t_s = MIN(t_s^Q, t_s^O), t_f = MAX(t_f^Q, t_f^O) \quad (29)$$

그림 8은 질의 Q와 객체 O의 존속구간이 [t_s^Q, t_f^Q], [t_s^O, t_f^O]라 할 때, 두 객체 간의 거리 제척을 나타낸다. 블록 부분은 두 객체가 동시에 존속하는 구간으로 [t_s^Q, t_f^Q]에 해당하며, 실제적 거리를 MBR 간의 거리로 구할 수 있다. 그러나 원으로 표시된 부분은 두 객체의 존속 구간이 서로 겹치지 않는 부분으로 질의에 대한 객체의 거리를 계산할 수 없다. 따라서 이러한 경우에는 MaxDist값을 가진다.

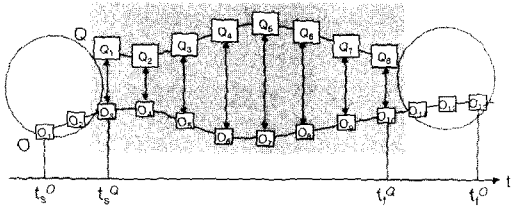


그림 8 MBR 사이의 거리 유사도

그림 9는 거리 유사도 검색을 수행하기 위한 `dis_search()` 알고리즘이다. `dis_search()`는 먼저 질의 객체 Q 에 대한 거리 변화를 먼저 계산한다. 식 (29)와 같이 질의에 대한 객체의 거리를 계산하기 위해 객체와 질의의 존속 구간을 비교하여 `create_interval()`을 통해 거리 유사도를 수행할 존속 구간을 설정한다. 존속 구간이 설정되면 `time_dist()`를 통해 각각의 시점에서 객체와 질의에 대한 거리를 계산한다. 이렇게 계산된 거리는 식 (29)와 같이 거리 유사도를 계산하기 위해 `object_dist()`를 수행한다.

```

Algorithm dis_search(Q, R)
Input : Q - 질의 객체, R - 검색 유사도
Output : O - 검색된 객체 또는 프레임의 리스트
{
    i:=0;
    calculate_dist(); //질의 객체에 대한 거리 변화량 계산
    while(검색할 객체가 존재할 때){
        calculate_interval(); //질의 객체와 검색 대상 객체의 존속 구간을 비교
        create_interval(); //검색을 수행할 구간을 생성
        while(검색 구간 동안){
            dist=time_dist(); // 검색 구간 내에서 질의 객체와 객체에 대한 거리 계산
            sum=sum+dist;
        }
        sim=object_dist(); // sum을 검색 구간으로 나누어 유사도를 계산
        if(sim>R){
            result[i]에 비교된 객체를 저장;
            i++;
        }
    }
    result[i]에 저장된 결과를 반환;
}
    
```

그림 9 거리 유사도 검색

4.4 궤적 유사도 검색

객체의 궤적 유사도 검색은 시간의 변화에 따른 객체의 공간적인 궤적의 변화에 대한 유사도 검색을 수행한다. 객체 계층은 객체의 공간적인 움직임을 파악할 수 있는 궤적 정보를 표현하였다. 궤적 유사도 검색은 객체를 직접 선택하고 선택된 객체의 궤적과 유사한 객체를 검색하거나 인터페이스 화면을 통해 스케치를 수행할 수 있다. 이를 통해 질의 Q 에 대한 객체의 궤적 유사도 검색을 수행한다. 궤적 유사도 검색에서 제공되는 질의 유형은 다음과 같다.

- 주어진 질의 Q 와 이동 방향이 유사한 객체 O 를 검색하라.

- 주어진 질의 Q 에 대해 이동 방향과 거리가 유사한 객체 O 를 검색하라.
- 주어진 질의 Q 와 이동 방향, 거리, 속도가 유사한 객체 O 를 검색하라.

궤적 유사도 검색을 수행하기 위해 질의 Q 의 궤적 T_Q 와 객체 O 의 궤적 T_O 가 아래와 같다고 하자.

$$T_Q = \langle (R_Q^1, V_Q^1, D_Q^1, I_Q^1), (R_Q^2, V_Q^2, D_Q^2, I_Q^2), \dots, (R_Q^m, V_Q^m, D_Q^m, I_Q^m) \rangle$$

$$\dots, (R_Q^m, V_Q^m, D_Q^m, I_Q^m) \rangle$$

$$T_O = \langle (R_O^1, V_O^1, D_O^1, I_O^1), (R_O^2, V_O^2, D_O^2, I_O^2), \dots, (R_O^n, V_O^n, D_O^n, I_O^n) \rangle$$

$$\dots, (R_O^n, V_O^n, D_O^n, I_O^n) \rangle$$

궤적 유사도에서는 이동 객체의 변화에 따라 객체의 이동 방향, 속도, 거리에 대한 상대적인 이동 변화를 통해 검색을 수행한다. 식 (30)은 질의 Q 의 궤적에 대한 객체 O 의 궤적 유사도 $ST(Q, O)$ 를 나타낸다. 이때, $Tra(Q, O)$ 는 질의와 객체의 궤적 차이로 식 (31)과 같다. $ST(Q, O)$ 는 0에서 1의 범위를 가지며, 유사할수록 1에 가깝다.

$$ST(Q, O) = 1 - Tra(Q, O) \tag{30}$$

$$Tra(Q, O) = \text{MIN} \sum_{j=1}^m \frac{RDV_{i,j+k}(Q, O)}{m}$$

$$(\forall k, 0 \leq k \leq n - m + 1) \tag{31}$$

$Tra(Q, O)$ 는 질의에 대한 객체의 이동 거리, 속도, 방향에 대한 거리 차를 계산한다. 질의 궤적의 각각의 공간적인 움직임에 대한 객체의 궤적 차이를 나타내는 $RDV_{i,j}(Q, O)$ 는 식 (32)와 같이 정의한다. 이때, ω_n 는 이동 방향, 거리, 속도에 부여된 가중치로 $\omega_n + \omega_e + \omega_d = 1$ 이 된다. $RDV_{i,j}(Q, O)$ 는 0에서 1의 범위를 가지며, 차가 적을수록 0에 가깝다. $RDV_{i,j}(Q, O)$ 는 질의 궤적의 움직임 각각에 대한 객체의 이동 거리 차 $RD(Q^i, O^j)$ 와 이동 방향 차 $RR(Q^i, O^j)$ 와 이동 속도의 차 $RV(Q^i, O^j)$ 을 통해 계산된다.

$$RDV_{i,j}(Q, O) = \omega_n RR(Q^i, O^j)$$

$$+ \omega_e RD(Q^i, O^j) + \omega_d RV(Q^i, O^j) \tag{32}$$

질의 궤적에 대한 객체의 이동 거리 차 $RD(Q^i, O^j)$ 는 질의 Q 의 i 번째 궤적 $\langle R_Q^i, V_Q^i, D_Q^i, I_Q^i \rangle$ 와 객체 O 의 j 번째 궤적 $\langle R_O^j, V_O^j, D_O^j, I_O^j \rangle$ 의 이동 거리를 나타내는 D_Q^i 와 D_O^j 의 이동 거리를 계산한다. 식 (33)은 질의에 대한 객체의 이동 거리 차 $RD(Q^i, O^j)$ 를 계산하는 식이다. 이때, $MaxD$ 는 질의에 대한 객체의 최대 이동 거리를 나타내며 응용에 따라 이 값은 변할 수 있다. 예를 들어, 축구 경기에서는 축구 경기장이 최대 거리가 된

다. $RD(Q^i, O^j)$ 는 0에서 1의 범위를 가지며 이동 거리 차가 없을수록 0에 가깝다.

$$RD(Q^i, O^j) = \frac{|D_Q^i - D_O^j|}{MaxD} \quad (33)$$

또한 질의 Q 에 대한 객체 O 의 속도 차를 나타내는 $RV(Q^i, O^j)$ 는 식 (34)와 같다. 질의 Q 의 i 번째 궤적의 속도가 V_Q^i 이고 객체 O 의 j 번째 궤적의 속도가 V_O^j 라 할 때, 질의에 대한 객체의 속도 차를 나타낸다. 이때, $MaxV$ 는 질의에 대한 객체의 최대 속도 차를 나타낸다. $RV(Q^i, O^j)$ 는 $RD(Q^i, O^j)$ 와 유사하게 0에서 1의 범위를 가지며 이동 속도 차가 없을수록 0에 가깝다.

$$RV(Q^i, O^j) = \frac{|V_Q^i - V_O^j|}{MaxV} \quad (34)$$

궤적 유사도 검색에서는 이동된 객체의 모양에 기반하여 검색을 수행한다. 제안하는 궤적 유사도 검색에서는 이동하는 객체의 전후 위치의 상대적인 방향 차를 이용하여 모양 기반 궤적 유사도 검색을 수행한다. 따라서 질의 Q 에 대한 객체 O 의 방향 차 $RR(Q^i, O^j)$ 은 식 (35)와 같다. 질의 Q 의 i 번째 궤적의 방향이 R_Q^i 이고 객체 O 의 j 번째 궤적의 방향이 R_O^j 라 할 때, 질의에 대한 객체의 방향 차를 나타낸다. 이때, $MaxR$ 는 질의에 대한 객체의 최대 방향 차로 나타낸다.

$$RR(Q^i, O^j) = \begin{cases} 0.5 + \frac{|RR_O^j - RR_Q^i|}{MaxR}, & RR_O^j RR_Q^i < 0 \\ \frac{|RR_O^j - RR_Q^i|}{MaxR}, & RR_O^j RR_Q^i > 0 \end{cases} \quad (35)$$

이 때,

$$|RR_O^j - RR_Q^i| = \begin{cases} |RR_O^j - RR_Q^i|, & |RR_O^j - RR_Q^i| < 180 \\ 360 - |RR_O^j - RR_Q^i|, & |RR_O^j - RR_Q^i| \geq 180 \end{cases}$$

이다. $RR(Q^i, O^j)$ 는 객체 간의 방향차로 동일한 방향으로 움직인 객체 사이에는 0에서 0.5의 범위로 나타내고 서로 다른 방향으로 움직인 객체 사이에는 가중치 0.5를 더하여 0.5에서 1의 범위를 갖도록 한다. 객체 O 의 궤적에 나타나는 방향의 변화가 $\langle R_O^1, R_O^2, \dots, R_O^n \rangle$ 일 때, 이전 객체의 위치에서 다음 위치로의 움직임 방향 차 RR_O^i 는 식 (36)과 같이 정의한다. 즉, 질의에 대한 객체의 이동 방향을 계산하기 전에 질의 또는 이동 객체 자체가 이전 위치에 대해 상대적인 이동 방향의 변화를 계산한다.

$$RR_O^i = \begin{cases} 180 - DR_O^i, & DR_O^i \geq 0 \\ -180 - DR_O^i, & DR_O^i < 0 \end{cases} \quad (36)$$

이 때, $RR_O^2 = DR_O^1$ 이고 $DR_O^i = \frac{R_O^i - R_O^{i-1}}{2} (2 \leq i \leq n)$

이다. RR_O^i 는 객체 O 의 i 번째의 이동에서 $i-1$ 번째의 이동에 대한 방향 변화를 나타내며 첫 번째 객체의 궤

적의 변화는 자신의 각도가 방향차가 된다. 단일 객체의 방향차 RR_O^i 는 $-180 \sim 180$ 의 범위를 갖는다.

그림 10은 궤적 유사도 Tra_Search()를 수행하는 알고리즘을 나타낸다. 질의 Q 에 대한 질의 궤적의 내용을 파악하기 위해 먼저 calculate_query()를 수행한다. 질의 궤적이 생성되면 object_relDist(), object_relDirect(), object_relVel()을 통해 질의에 대한 객체의 방향 차, 속도 차, 거리 차를 계산하고 그 합을 계산한다. 이렇게 계산된 질의와 객체의 궤적 차를 이용하여 식 (34)에 대한 궤적 유사도를 계산하기 위해 object_tra()를 계산한다.

```

Algorithm Tra_Search(Q, W, R)
Input : Q = 질의 객체, W = 가중치, R = 검색 유사도
Output : O = 검색된 객체
{
  i=0;
  calculate_query(); // 질의에 대한 궤적 변화를 계산
  while(검색할 객체가 존재할 때){
    while(객체의 궤적 변화 동안){
      dist[i]=object_relDist(); // 특정 시점의 거리 차
      direct[i]=object_relDirect(); // 특정 시점의 방향 차
      vel[i]=object_relVel(); // 특정 시점의 속도 차
      sum[i]=sum_dist(); // 가중치를 고려하여 특정 시점의 궤적 차를 계산
    }
    sim=object_tra(); // sum[i]를 이용하여 궤적 유사도를 계산
    if(sim ≥ R){
      result[i]에 비교된 객체를 저장;
      i++;
    }
  }
  result[i]에 저장된 결과를 반환;
}
    
```

그림 10 궤적 유사도 검색 알고리즘

4.5 가중치 혼합 유사도 검색

제안하는 모델링에서는 이동 객체 간의 시각적 특징, 거리, 궤적에 사용자가 가중치를 부여함으로써 가중치 혼합 유사성 검색이 가능하다. 이러한 질의를 수행하기 위해 혼합 유사도를 계산하기 위한 정의가 필요하다. 질의 Q 에 대한 객체 O 의 가중치 혼합 검색 유사도 $SS(Q, O)$ 는 식 (37)과 같이 정의한다. $SS(Q, O)$ 는 식 (33), 식 (34), 식 (35)에서 정의한 시각적 특징 유사도 $ST(Q, O)$ 와 거리의 유사도 $SD(Q, O)$ 그리고 궤적의 유사도 $RD(Q, O)$ 에 부여된 가중치 w_{ss} 를 이용하여 계산한다. 이때, $w_{st} + w_{sd} + w_{ss} = 1$ 이다. $SS(Q, O)$ 는 0~1의 범위를 갖으며 유사도가 좋을수록 1에 가깝다.

$$SS(Q, O) = w_{st}SC(Q, O) + w_{sd}SD(Q, O) + w_{ss}ST(Q, O) \quad (37)$$

5. 실험 및 비교 분석

5.1 실험 및 사용자 인터페이스

본 논문에서 제안하는 이동 객체에 대한 검색을 수행

하기 위해 128MB 메모리를 갖는 펜티엄-III 700 MHz 의 Windows 2000 Server 운영체제에서 MS SQL Server 2000을 이용하였다. 구현 언어로는 Microsoft Visual C++를 사용하였다. 농구 비디오 데이터는 경기장이라는 한정된 공간을 배경으로 짧은 시간에 많은 이동을 가지는 객체를 포함하고 있어서 이동 객체의 궤적을 추출하는데 용이하다. 따라서, 실험 데이터는 공간적인 위치를 변화를 쉽게 파악할 수 있는 실제 농구 비디오 데이터에서 사용한다.

그림 11은 비디오 데이터베이스에서 이동 객체 검색을 위한 시스템 구조를 나타낸 것이다. 이동 객체 검색을 위한 시스템은 데이터 처리(Data Processing), 질의 처리(Query Processing) 그리고 데이터베이스(Database)의 세 부분으로 구성되어 있다. 데이터 처리는 이동 객체의 내용을 추출하기 위한 부분으로 키프레임을 추출하는 키프레임 추출기(Keyframe Extractor)와 키프레임을 통해 내용 정보들을 생성하는 내용 추출기(Contents Extractor)로 구성되어 있다. 질의 처리는 사용자의 질의 요청에 대한 내용을 처리하고 그 결과를 전송하기 위한 부분으로 사용자 인터페이스와 질의 처리기 부분으로 구성되어 있다. 데이터베이스에는 의미적 내용을 표현하기 위해 필요한 의미적 지식 데이터베이스(Semantic Knowledge Database) 부분과 실제적인 이동 객체의 내용을 저장하는 이동 객체 데이터베이스(Moving Object Database)로 구성되어 있다.

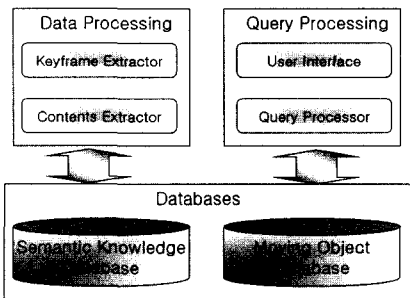


그림 11 시스템 구조

실험을 위해 농구 비디오에서 1초 단위로 키프레임을 추출하고 10명의 선수에 대한 이동과 의미적 내용을 구성한다. 각각 이동 객체에 해당하는 운동 선수들은 각각 5~15개의 궤적 정보를 포함하고 있다. 또한 질의 데이터 10개 생성하고 각 질의에 대한 5~15개의 이동 정보들로 구성한다. 실제로 모든 객체는 3차원 공간에서 존재하지만 비디오는 2차원 평면으로 보여지기 때문에 객체는 2차원으로 정사영하여 표현한다. 이동 객체의 의미적 내용은 농구에 발생할 수 있는 각종 사건과 운동 선

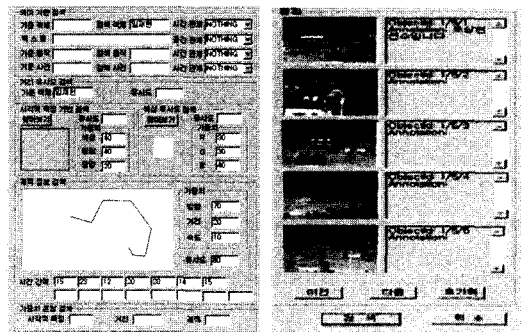
표 1 실험 데이터

특성	값
포맷 형식	avi 파일
비디오 시간	54 분
파일 크기	544 MB
초당 프레임 수	23개

수들의 행위를 수작업을 통해 추출한다. 실험에서 사용된 비디오 데이터의 특징은 표 1과 같다.

그림 12는 이동 객체 검색을 위한 사용자 인터페이스 및 검색 결과 화면을 나타낸 것이다. 그림 12의 (a)는 사용자 인터페이스를 나타낸 것으로 4절에서 언급한 각각의 검색을 지원하기 위한 부분과 가중치를 부여하는 부분으로 구성되어 있다. 사용자에게 의해 검색을 수행한 결과를 나타내는 그림 12의 (b)는 검색 결과에 대한 키프레임과 내용 정보들을 함께 나타낸다. 사용자의 검색 인터페이스에 존재하는 의미 기반 검색에는 특정 객체를 기준으로 연관 관계성에 기반한 검색을 수행하기 위한 “기준 객체” 또는 실제적인 검색 대상이 되는 “검색 객체”를 선택할 수 있다. 또한 이러한 객체의 선택에 따라 검색 조건이 되는 동작이나 사건 등을 선택할 수 있다. 거리 유사도 검색을 위해서는 검색의 기준이 되는 “기준 객체”와 유사도를 입력한다. 시각적 특징 기반 검색을 수행하기 위해서는 “찾아보기”를 통해 이미지 목록에서 특정 이미지를 선택하고 시각적 특징 유사도를 입력한다. 궤적 유사도 검색은 사용자가 직접 검색 인터페이스에 직접 스케치를 수행하고 유사도를 입력하여 수행할 수 있다.

사용자 인터페이스를 통해 질의 처리를 수행하기 위해 먼저 검색 인터페이스를 통해 사용자의 질의가 입력되면 입력된 질의의 유형을 분석한다. 분석된 질의 유형을 통해 질의 처리기는 입력 조건에 따라 데이터베이스에 질의 요청을 하기 위한 SQL문을 생성하고 이를 데



(a) 검색 인터페이스 (b) 검색 결과

그림 12 사용자 인터페이스

이타베이스 시스템에 전송한다. 데이터베이스 시스템은 이동 객체 모델링에 따라 구성된 데이터베이스를 SQL 문을 수행한 질의 결과를 검색 결과 인터페이스 출력한다. 질의 결과 인터페이스는 질의 결과에 따라 검색 결과 인터페이스의 왼쪽에는 객체의 시각적 이미지를 출력하고 오른쪽에는 객체의 텍스트 정보를 출력한다. 검색 결과 인터페이스를 통해 출력된 이미지를 더블 클릭하면 검색 결과 인터페이스에 나타난 내용을 재생할 수 있다.

5.2 비교 분석

본 논문에서는 제안하는 이동 객체 모델링 및 검색 기법의 우수성을 보이기 위해 제공되는 질의 유형과 객체의 표현 기법에서 비교를 수행한다. 제안하는 기법과 비교를 수행하기 위해 편의상 [9,11,12,14]에서 제안된 기법을 MMO, STRR, MQV, MRMO라 한다. 또한 본 논문에서 제안하는 기법을 CBRO(Content Based Retrieval of moving Objects)라 한다. 표 2는 기존에 연구된 이동 객체 검색과 제안하는 기법을 비교 분석한 결과이다. 표 2에서 알 수 있듯이 기존에 제안된 이동 객체 모델링 및 검색 기법들은 객체의 공간적인 이동 궤적에 대한 검색에 초점을 맞추고 있다. 따라서 공간적인 변화 이외에 의미적 내용에 대한 변화 또는 시각적 특징의 변화 등을 검색하지 못한다. MQV는 시각적 특징에 기반한 검색을 지원하지만 공간적인 위치 변화에 따른 궤적 검색을 지원하지 못한다. 또한 제안하는 기법에서는 궤적 검색 과정에서 단순히 객체와 궤적의 이동 방향 또는 거리의 차를 이용한 것이 아니라 객체의 이전 움직임과 이후 움직임의 상대적 차를 이용하여 검색한다.

제안하는 검색을 수행하기 위해서는 다양한 영상 처리 기술을 이용하여 객체의 특징을 추출하고 추출된 특징들 사이의 관계를 연결하기 위한 매핑 기법들이 필요하다. 영상 처리 기술 기술들은 다양한 검색 기법을 지원하기 위한 반드시 필요한 기술들이며 이러한 영상 처리 기술들이 영상 처리를 전문적으로 연구하는 분야에서 별도의 연구로 수행되고 있다. 본 논문에서 제안하는 모델링 및 검색을 수행하기 위해서는 영상 처리 기술에

의해 많은 영향을 받을 수 있으며 영상 처리 기법들이 제대로 수행되지 못할 경우 모델의 효율성 및 검색의 성능을 저하시킬 수 있다. 그러나 최근 영상 처리를 전문적으로 연구하는 분야에서 이러한 연구들이 진행되고 있기 때문에 최신의 영상 처리 기술을 도입할 경우 이러한 문제는 해결될 수 있다.

6. 결론 및 향후 연구

본 논문에서는 비디오 데이터에 포함된 이동 객체에 대한 내용 기반 검색을 위한 다양한 검색 유형을 지원하기 위한 모델링 및 연산자를 제안하였다. 본 논문에서는 이동 객체의 시각적인 특징, 공간적인 위치 변화뿐만 아니라 이동 객체와 연관된 의미적 내용을 표현하였다. 또한 새로운 모델링 기법에 의해 표현된 각각의 내용들을 효과적으로 검색하기 위한 연산자를 제안한다. 이를 기반으로 제안된 모델링에 표현된 내용에 대한 검색을 수행하기 위해 제안된 연산자들을 이용하여 검색을 수행하는 과정을 기술하였다. 본 논문에서 제안된 기법들은 기존의 이동 객체 검색에서 수행되지 못한 의미적 내용 검색, 시각적 특징 기반의 검색, 거리 유사도 검색, 궤적 유사도 검색을 수행할 수 있으며 새로운 연산자를 통해 보다 효과적인 질의 수행을 지원할 수 있다.

향후 연구 방향으로 다양한 데이터 크기에 따른 검색 시간에 대한 비교 분석과 질의 처리 시간 최적화에 대한 연구를 수행할 예정이다. 또한 이동 객체에 시공간 관계성 및 이에 대한 검색 기법에 대한 연구를 수행할 예정이다.

참고 문헌

[1] R. Hjelmsvold, "VideoSTAR - A Database for Video Information Sharing," Dr.Ing. Thesis, Norwegian Institute of Technology, November 1995.
 [2] M. La Cascia and E. Ardizzone, "JACOB : Just a content-based query system for video databases," Pro. IEEE International Conference On Acoustics, Speech and Signal Processing, pp.7-10, 1996.
 [3] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram and D. Zhong, "VideoQ : An Automatic Content-Based Video Search System Using Visual Cues,"

표 2 모델링 기법들의 비교

비교항목	비교대상	MNO	STRR	MQV	MRMO	CBRO
시각적 특징		×	×	○	×	○
궤적		방향	방향, 거리	×	방향, 거리	방향, 거리, 속도
의미적 내용		×	×	×	×	동작, 사건
키프레임		×	×	○	×	○
거리 유사도		×	×	×	×	○

- Pro. ACM Multimedia, pp.313-324, 1997.
- [4] M. Petkovic and W. Jonker, "A Framework for Video Modeling," Pro. Eighteenth IASTED International Conference Applied Informatics, 2000.
 - [5] J. L. Koh, C. S. Lee and A. L. P. Chen, "Semantic Video Model for Content-Based Retrieval," Proc. IEEE International Conference on Multimedia Computing and Systems, Vol.2, pp.472-478, 1999.
 - [6] D. Papadias and Y. Theodoridis, "Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures," International Journal of Geographic Information Science, Vol.11, No.2, pp.111-138, 1997.
 - [7] J. Z. Li, M. T. Ozsü and D. Szafron. "Modeling of Video Spatial Relationships in an Object Oriented Database Management System," Pro. International Workshop on Multimedia DBMS, pp.124-132, 1996.
 - [8] M. E. Donderler, O. Ulusoy and U. Gudukbay, "A Rule-based Approach to Represent Spatio-Temporal Relations in Video Data," Pro. International Conference on Advances in Information Systems, vol.1909, 2000.
 - [9] J. Z. Li, M.T. Ozsü and D. Szafron, "Modeling of Moving Objects in a Video Database," Pro. IEEE International Conference on Multimedia Computing and Systems, pp.336-343, 1997.
 - [10] M. Nabil, A. H. H. Ngu, and J. Shepherd, "Modeling moving objects in multimedia databases," Pro. Fifth International Conference on Database Systems for Advanced Applications, pp.67-76, 1997.
 - [11] C. B. Shim and J. W. Chang, "Spatio-temporal representation and retrieval using moving object's trajectories," ACM Multimedia Workshops 2000, pp.209-212, 2000.
 - [12] Z. Aghbari, K. Kaneko and A. Makinouchi, "Modeling and Querying Videos by Content Trajectories," Pro. IEEE International Conference on Multimedia and Expo, pp.463-466, 2000.
 - [13] Z. Aghbari, K. Kaneko and A. Makinouchi, "VST Model: A Uniform Topological Modeling of the Visual-Spatio-Temporal Video Features," Pro. IEEE International Conference on Multimedia Computing and Systems 99, Vol.2, pp.163-168, 1999.
 - [14] M. Nabil, A. H. H. Ngu and J. Shepherd, "Modeling and Retrieval of Moving Objects," Multimedia Tools and Applications Vol.13, No. 1, pp.35-71, 2001.



복 경 수

1998년 2월 충북대학교 수학과 이학사.
2000년 2월 충북대학교 정보통신공학과
공학석사. 2000년 2월~현재 충북대학교
정보통신공학과 박사과정. 관심분야는 자
료 저장 시스템, 멀티미디어 데이터베이
스, 이동객체 데이터베이스, 고차원 색인

구조, 시공간 색인구조 등

유 재 수

정보과학회논문지 : 데이터베이스
제 31 권 제 1 호 참조