이동 객체의 내용 및 개념 기반 검색을 위한 시공간 모델링에 근거한 시그니쳐 기반 비디오 색인 기법

심 춘 보† · 장 재 우††

요 약

본 논문에서는 비디오 데이터가 지니는 이동 객체의 궤적(Moving Object's Trajectory)을 효과적으로 모델링할 수 있는 시공간 표현 기법(Spatio-Temporal Representation Scheme)과 궤적을 이용한 사용자 질의에 대해 효율적인 검색을 위한 새로운 시그니쳐 기반 접근 기법을 제안한다. 제안하는 시공간 표현 기법은 궤적을 기반으로 하는 내용 기반 검색(Content-based Retrieval)과 궤적에서 일어나는 위치 정보를 통해 얻어진 개념(의미)을 이용한 개념 기반 검색(Concept-based Retrieval)을 지원한다. 아울러, 제안하는 시그니쳐 기반 접근 기법은 데이터 파일을 직접 접근하기 전에 전체 시그니쳐들을 탐색하여 필터링을 수행한 후, 검색된 후보 시그니쳐들에 대해서만 디스크를 접근하기 때문에 순차 탐색에 비해 많은 수의 디스크 접근 횟수를 감소시킴으로써 검색 성능을 향상시킨다. 마지막으로, 성능 평가를 통해 제안하는 방법이 검색 효과(Retrieval Effectiveness) 및 효율(Retrieval Efficiency) 측면에서 기존의 방법인 Li나 Shan의 방법에 비해 우수함을 보인다.

A Signature-based Video Indexing Scheme using Spatio-Temporal Modeling for Content-based and Concept-based Retrieval on Moving Objects

Choon-Bo Shim† · Jae-Woo Chang††

ABSTRACT

In this paper, we propose a new spatio-temporal representation scheme which can model moving objects' trajectories effectively in video data and a new signature-based access method for moving objects' trajectories which can support efficient retrieval on user query based on moving objects' trajectories. The proposed spatio-temporal representation scheme supports content-based retrieval based on moving objects' trajectories and concept-based retrieval based on concepts(semantics) which are acquired through the location information of moving object's trajectories. Also, compared with the sequential search, our signature-based access method can improve retrieval performance by reducing a large number of disk accesses because it access disk using only retrieved candidate signatures after it first scans all signatures and performs filtering before accessing the data file. Finally, we show the experimental results that proposed scheme is superior to the Li and Shan's scheme in terms of both retrieval effectiveness and efficiency.

키워드 : 이동 객체 궤적(moving object trajectories), 시공간 표현 기법(spatio-temporal representation scheme), 내용 및 개념 기반 검색(content-and concept-based retrieval)

1. 서 론

최근 들어, 멀티미디어 데이터베이스 응용에서 내용 기반 검색 및 유사성에 기반한 검색에 대한 관심이 날로 증가하고 있다. 특히, 대용량의 멀티미디어 데이터베이스에서 사용자의 다양한 요구를 보다 효과적으로 처리하기 위해서는 효율적인 내용 기반 멀티미디어 검색 기법과 그에 따른 유사성 기반 검색이 요구된다. 실질적으로, 멀티미디어 데이터 중에서도 텍스트나 이미지 데이터와는 달리 비디오 데이터가 지니는 가장 중요한 특징은 이동 객체에 대한 움직임 정보이다. 이러한 움직임 정보는 각각의 프레임 내에서의 객체들간의 공간적인 정보와 일련의 프레임들간의 시간적인 정보가 결합된 시공간 관계성을 통해 표현될 수 있으며, 비디오 데이터에 대한 사용자의 내용 및 개념 기반 검색을 수행하는 데 있어 매우 중요한 역할을 한다. 비디오 데이터베이스에서 이동 객체의 궤적을 이용한 내용 기반 검색 질의로는 다음과 같은 질의가 가능하다. "*사용자 인터페이스를 통해 스케치된 이동 객체의 궤적과 유사한 궤적을 가진 이동 객체를 포함하는 모든 비디오 샷(Shot)을 찾아라.*" 또한 응용 분야에 따라 일련의 이동

† 준 회 원 : 전북대학교 컴퓨터공학과
†† 종신회원 : 전북대학교 컴퓨터공학과 교수, 전자정보신기술연구센터
논문접수 : 2001년 8월 29일, 심사완료 : 2001년 11월 23일

객체의 궤적으로부터 어떤 개념(Concept) 혹은 의미(Semantic) 정보를 유추해 낼 수 있다. 가령, 축구 비디오 데이터베이스의 경우 일련의 축구공의 궤적으로부터 코너킥(Corner Kick), 패널티 킥(Penalty Kick) 등과 같은 개념(의미) 정보를 대략적으로 얻을 수 있다. 따라서, 사용자는 비디오 데이터에서 이동 객체의 궤적을 이용한 내용 기반 검색과 더불어 다수의 궤적들로부터 얻어낸 개념을 통한 개념 기반 검색을 수행할 수 있다.

한편, 내용 기반 비디오 검색에 관한 초기의 대부분의 연구[1-6]는 비디오 데이터로부터 추출된 의미 및 내용 정보를 효율적으로 모델링 할 수 있는 데이터 표현 기법과 더불어 다른 미디어에 비해 큰 용량을 차지하는 비디오 데이터를 분해해서 데이터베이스에 효과적으로 저장하는 데이터 저장 기법에 큰 관심을 가져왔다. 그러나, 점점 대용량의 멀티미디어 데이터를 다루는 데 있어 연구의 주된 관심은 사용자의 다양한 질의를 어떻게 적절히 분석해서 분석된 결과를 토대로 빠른 검색 성능을 제공할 수 있는 검색의 효율성에 있다.

따라서, 본 논문에서는 비디오 데이터가 지니는 이동 객체의 궤적(Moving Object's Trajectory)[7-10]을 효율적으로 모델링 할 수 있는 새로운 시공간 표현 기법(Spatio-Temporal Representation Scheme)과 궤적을 이용한 다양한 사용자 질의에 대해 효율적인 검색을 위한 시그니쳐 기반 접근 기법(Signature-based Access method for Moving Objects' Trajectories, SAMOT)을 제안한다. 제안하는 시공간 표현 기법은 궤적을 기반으로 하는 내용 기반 검색(Content-based Retrieval)과 궤적이 일어나는 위치 정보를 통해 얻어진 개념(의미)을 이용한 개념 기반 검색(Concept-based Retrieval)을 지원한다. 또한 제안하는 시그니쳐 기반 접근 기법은 데이터 파일을 직접 접근하기 전에 전체 시그니쳐[11]들을 탐색하여 필터링을 수행하기 때문에 순차 탐색에 비해 많은 수의 디스크 접근 횟수를 감소시킴으로써 검색 성능을 향상시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 이동 객체를 이용한 비디오 검색에 관한 관련 연구를 분석한다. 3장에서는 본 연구에서 제안하는 내용 및 개념 기반 비디오 검색을 지원하는 시공간 표현 기법을 설명한다. 4장에서는 제안하는 시공간 표현 기법을 통해 축구 비디오 데이터베이스에서의 내용 및 개념 기반 검색을 제시한다. 5장에서는 검색의 효율성을 위해 제안하는 시그니쳐 기반 접근 기법에 대해서 기술한다. 6장에서는 제안하는 방법과 타 연구와 성능 비교를 수행한다. 마지막으로, 7장에서 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

이미지 내의 객체들 간의 공간 관계성을 이용한 내용 기반 이미지 검색 기술에 관한 연구는 현재 상당한 연구 결과를 보이고 있다. 그러나, 비디오나 애니메이션을 기반으로 이동 객체의 시공간 관계성을 이용한 내용 기반 비디오 검색에 관한 연구들은 현재 진행 중에 있다. 따라서, 여기서는 본 논문과 밀접한 관련이 있는 두 가지 연구에 대해서 기술한다.

첫째, John Z. Li(이하 Li)[7, 8]은 어떤 일정시간 동안 객체의 위치가 변하는 객체를 이동 객체(moving object)로 간주하고 이에 대해서 8개의 방향 즉, North(NT), NorthWest (NW), West(WT), SouthWest(SW), South(ST), SouthEast (SE), East(ET), NorthEast(NE)을 고려하여 객체의 궤적을 표현하고 있다. 임의의 시간 간격 I_i 동안 객체 A의 동작(motion)은 (S_i, d_i, I_i)로 표현하며, 여기에서, S_i는 객체 A의 변위(displacement)이고 d_i는 객체 A의 이동 방향(direction)을 의미한다. 따라서, 어떤 일련의 주어진 시간 간격 $<I_1, I_2, \cdots, I_n>$에 대해서 객체 A의 궤적(trajectory)는 다음의 일련의 동작들로 표현한다.

$$<(S_1, d_1, I_1), (S_2, d_2, I_2), \cdots, (S_n, d_n, I_n)>$$

시간 간격 I_k 동안 이동 객체 A와 B사이의 시공간 관계성은 $A(\alpha, \beta, I_k)B$로 표현하며, 여기에서, α는 객체 A와 B사이의 위상 관계(topological relation) 즉, DJ(DisJoint), TC (TouCh), EQ(EQual), IN(INside), CB(Covered_By), CT(ConTains), CV(CoVers), OL(OverLap) 중의 하나를 의미하고, β는 객체 A와 B의 방향 관계(directional relation)를 나타낸다. 따라서, 객체 A와 B사이의 시공간 관계성은 다음의 일련의 동작들로 표현한다.

$$<(\alpha_1, \beta_1, I_1), (\alpha_2, \beta_2, I_2), \cdots, (\alpha_n, \beta_n, I_n)>$$

Li는 <표 1>과 같이 위상 관계에 대한 거리(distance)와 방향 관계에 대한 거리를 이용하여 이동 객체 A의 궤적과 객체 A와 객체 B사이의 시공간 관계성에 대한 유사도(Similarity)를 계산하였다. 객체 A의 궤적 = $<M_1, M_2, \cdots, M_m>$과 객체 B의 궤적 = $<N_1, N_2, \cdots, N_n>$ 사이의 유사도 함수는 TrajSim(A, B)는 다음과 같다.

$$\min Diff(A, B) = MIN \sum_{i=1}^{m} distance(M_i, N_{i+j}) \quad (\forall j,\ 0 \le j \le n-m)$$

$$TrjaSim(A, B) = \frac{\max Diff(A, B) - \min Diff(A, B)}{\max Diff(A, B)}$$

여기서, M_i는 객체 A 궤적의 i번째 동작을 의미하며, m과 n은 각각 객체 A와 B의 궤적을 이루는 동작의 수를 나타낸다. 그리고 maxDiff(A, B)는 객체 A와 B의 궤적 사이의 거리의 최대 차를 의미하며 4*m으로 표현된다.

〈표 1〉 방향과 위상 관계에 대한 거리

(a) 방향 관계에 대한 거리

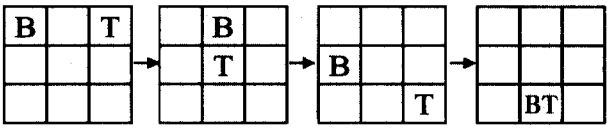
	NT	NW	NE	WT	SW	ET	SE	ST
NT	0	1	1	2	3	2	3	4
NW	1	0	2	1	2	3	4	3
NE	1	2	0	3	4	1	2	3
WT	2	1	3	0	1	4	3	2
SW	3	2	4	1	0	3	2	1
ET	2	3	1	4	3	0	1	2
SE	3	4	2	3	2	1	0	1
ST	4	3	3	2	1	2	1	0

(b) 위상 관계에 대한 거리

	DJ	TC	EQ	IN	CB	CT	CV	OL
DJ	0	1	6	4	5	4	5	4
TC	1	0	5	5	4	5	4	3
EQ	6	5	0	4	3	4	3	6
IN	4	5	4	0	1	6	7	4
CB	5	4	3	1	0	7	6	3
CT	4	5	4	6	7	0	1	4
CV	5	4	3	7	6	1	0	3
OL	4	3	6	4	3	4	3	0

둘째, Shan[9]는 내용 기반 비디오 검색을 위해 단일 동작 궤적(single motion trajectory)과 다중 동작 궤적(multiple motion trajectory)를 이용한 유사 검색(similarity retrieval) 알고리즘을 제시하였다. 먼저, 단일 동작 궤적(single motion trajectory)을 이용한 검색을 위해, 객체의 궤적(trajectory)은 일련의 세그먼트(segment)의 집합으로 나타내고, 각각의 세그먼트는 0°~360°까지의 각도로서 표현한다. 사용자 질의 궤적과 데이터베이스의 궤적 사이의 유사성을 계산하기 위해 OCM(Optimal Consecutive Mapping)과 OCMR(Optimal Consecutive Mapping with Replication)이라는 두 가지의 유사성 측정 알고리즘을 제안하였으며, 이 알고리즘은 모두 단일 동작 궤적의 방향 정보만을 이용하여 유사성을 계산한다. 그리고, 다중 동작 궤적을 이용한 검색을 위해서는 단순히 기존의 2D 이미지 내의 객체간의 공간 관계성을 위해 Chang[12]에 의해서 제안된 2D 스트링 방법(2D String Scheme)을 이용하였다. 따라서, 다중 동작 궤적은 일련의 심볼(symbol) 객체로 나타내며, 각각의 심볼 객체는 2D 스트링으로 표현한다. 예를 들어, (그림 1)은 *"마주보고 달려오는 버스와 트럭이 서로 부딪치는 비디오 장면(샷)"*으로 *"버스(B)"*과 *"트럭(T)"* 두 개의 궤적으로 구성된다. (그림 2)는 (그림 1)과 같은 다중 동작 궤적을 2D 스트링으로 표현한 예이다. 그러나, Shan 기법은 궤적을 모델링하는 있어 단순히 방향정보만을 이용하고 있으며, 시간 정보에 대해서는 고려하고 있지 않으며, 또한 다중 동작 궤적을 위해서는 2D 스트링을 이용하여 공간 관계를 표현하기 때문에 보다 복잡하고 정확한 공간 관계를 요구하는 응용 분야에는 적합하지 않다.

(그림 1) 다중 동작 궤적의 예



B : Bus T : Truck

(그림 2) 2D 스트링(2D String)으로 표현한 예

3. 내용 및 개념 기반 검색을 위한 새로운 시공간 표현 기법

비디오 데이터는 이미지 데이터와는 달리 공간 정보와 시간 정보를 모두 포함하고 있기 때문에, 이동 객체를 보다 효율적으로 표현하기 위해서는 이동 객체에 대한 공간 관계성(Spatial Relationship)[13]과 시간 관계성(Temporal Relationship)[14]을 모두 고려해야 한다. 따라서, 본 논문에서는 하나의 객체에 대한 궤적을 나타내는 단일 궤적(Single Trajectory : ST)와 둘 이상의 다수의 객체에 대한 궤적을 나타내는 다중 궤적(Multiple Trajectories : MT)를 위한 새로운 시공간 표현 기법을 제안한다. 아울러, 제안하는 시공간 표현 기법은 사용자에게 보다 빠른 검색 결과를 제공할 수 근사 탐색(Approximation Search)을 지원하며, 사용자 인터페이스(GUI)를 이용해 사용자가 직접 스케치한 궤적을 기반으로 하는 내용 기반 검색과 이동 객체의 일련의 다수의 궤적들과 궤적이 일어나는 위치 정보를 통해 얻어진 개념(의미)을 이용한 개념 기반 검색을 지원한다. 또한 개념 기반 검색 가운데 이동 객체의 행위자(Actor)를 기반으로 하는 행위자 기반 검색을 지원한다. 즉, 축구 비디오 데이터의 경우 *"A라는 선수가 슈팅을 하는 궤적을 가진 모든 축구 비디오 샷을 찾아라"*와 같은 질의를 지원한다.

3.1 단일 궤적(Single Trajectory : ST)

[정의 1] 객체 A를 이동 객체라고 할 때, 이동 객체 A의 단일 궤적, $ST^{(A)}$는 다음과 같이 IE와 ME로 구성된다.

$$ST^{(A)} = IE +^{1)} ME$$

IE(Instantaneous Element)는 궤적을 구성하는 전체 시점

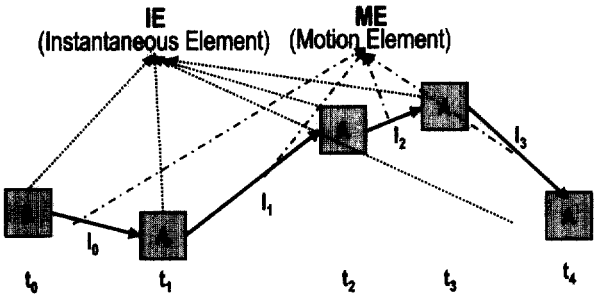
1) 본 논문에서 사용하는 '+' 연산자는 산술 연산자가 아닌 연결(concatenation) 연산자를 의미한다.

들 즉, t_0에서 t_n까지의 궤적을 이루는 각 시점에서의 정보를 나타내며 $P_n^{(A)} = \{P_0^{(A)}, P_1^{(A)}, \cdots, P_n^{(A)}\}$로 표현한다. P_i는 (L_i, A_i)로 구성되며 여기서, L_i는 움직임 객체 A의 위치 정보(Location)를 의미한다. 위치 정보는 일반적인 위치 정보 즉 좌표 정보가 될 수 있고, 본 논문에서 제안하는 개념 기반 검색을 위해 정의된 새로운 개념 기반 위치 정보가 될 수 있다.(개념 기반 위치 정보에 대해서는 다음 장을 참조). 그리고 A_i는 해당 위치에서의 이동 객체를 소유하거나 혹은 관련 있는 행위자(Actor)를 의미한다. 한편, ME(Motion Element)는 궤적을 구성하는 각 동작(Motion)을 나타내며 $\{M_{n-1}\} = \{M_0, M_1, \cdots, M_{n-1}\}$으로 표현한다. M_i는 (α_i, D_i, I_i)로 구성되며, α_i는 시간 간격 I_i 동안 움직인 방향으로서 0°~360°까지의 실제 각도(Real Angle)로서 나타내며, D_i는 시간 간격 I_i 동안 움직인 거리(Distance)를 의미한다. I_i는 움직임이 일어나는 시작 시점(t_i)과 끝 시점(t_{i+1})으로 구성된 시간 간격을 나타낸다. 따라서, 이동 객체 A를 위한 $ST^{(A)}$는 다음과 같으며, (그림 3)에 나타난다.

$$ST^{(A)} = \{P_n^{(A)}\} + \{M_{n-1}\}$$

아울러, 주어진 시간 간격 리스트 $\{I_0, I_1, \cdots, I_{n-1}\}$에 대해서, 이동 객체 A의 단일 궤적 $ST^{(A)}$는 다음과 같이 표현된다.

$$\begin{aligned} ST^{(A)} &= \{P_0^{(A)}, P_1^{(A)}, \cdots, P_n^{(A)}\} + \{M_0, M_1, \cdots, M_{n-1}\} \\ &= \{(L_0^{(A)}, A_0^{(A)}), (L_1^{(A)}, A_1^{(A)}), \cdots, (L_n^{(A)}, A_n^{(A)})\} \\ &\quad + \{(\alpha_0, D_0, I_0), (\alpha_1, D_1, I_1), \cdots, (\alpha_{n-1}, D_{n-1}, I_{n-1})\} \end{aligned}$$



(그림 3) 움직임 객체 A의 단일 움직임 경로

3.2 다중 궤적(Multiple Trajectory : MT)

본 논문에서는 둘 이상의 이동 객체로 이루어진 궤적을 다중 궤적(Multiple Trajectory : MT)라고 명명한다. 그러나, 실제로 둘 이상의 이동 객체로 이루어진 궤적은 단지, 두 객체(객체 A, 객체 B)로만 이루어진 궤적의 조합으로 표현이 가능하기 때문에, 본 논문에서는 두 객체로만 이루어진 궤적을 관계 궤적(Relationship Trajectory : RT)로 명명한다.

[정의 2] 객체 A 또는 객체 B를 이동 객체라고 할 때, 이동 객체 A와 B 사이의 궤적을 관계 궤적, $RT^{(A,B)}$라고 하며 다음과 같다.

$$RT^{(A,B)} = IE + ME$$

IE(Instantaneous Element)는 단일 궤적에서와 마찬가지로 궤적을 구성하는 전체 시점들 즉, t_0에서 t_n까지의 궤적을 이루는 각 시점에서의 정보를 나타내며 $\{P_n^{(A,B)}\} = \{P_0^{(A,B)}, P_1^{(A,B)}, \cdots, P_n^{(A,B)}\}$으로 표현한다. 여기서, $P_i^{(A,B)} = (L_i^{(A)}, A_i^{(A)}, L_i^{(B)}, A_i^{(B)}, R_i, \alpha_i)$로 이루어져 있으며, $L_i^{(A)}$와 $A_i^{(A)}$는 각각 이동 객체 A의 위치 정보와 행위자를 의미하며, $L_i^{(B)}$와 $A_i^{(B)}$는 이동 객체 B의 위치 정보와 행위자를 나타낸다. 그리고 R_i은 이동 객체 A와 B사이의 위상 관계성 즉, FA(FarAway), DJ(DisJoint), MT(MeeT), OL(OverLap), CL(Is inCLuded by), IN(INclude), SA(SAme) 중에 하나의 값으로 표현한다. 그리고 α_i는 이동 객체 A(기준 객체)에 대한 객체 B(대상 객체)의 방향 관계성을 나타낸다. 한편, ME(Motion Element)는 궤적을 구성하는 각 동작(Motion)을 나타내며 $\{M_{n-1}\} = \{M_0, M_1, \cdots, M_{n-1}\}$으로 표시한다. M_i는 (D_i, I_i)로 구성되며, D_i는 시간 간격(I_i) 동안 객체 A에 대한 객체 B의 상대적인 이동 거리(Relative Moving Distance)로 0에서 100사이의 정규화된 값으로 나타낸다. 즉, 객체 A의 이동 거리와 객체 B의 이동 거리가 같을 때 D_i를 50으로 정하고, 이를 기준으로 객체 A의 이동 거리가 객체 B의 이동 거리보다 클 경우에는 50에서 100 사이의 값을 갖으며, 그와 반대일 경우에는 0에서 50사이의 값을 갖는다. 그리고, I_i는 단일 궤적에서와 같은 의미이다. 따라서, $RT^{(A,B)}$는 다음과 같다.

$$RT^{(A,B)} = \{P_n^{(A,B)}\} + \{M_{n-1}\}$$

주어진 시간 간격 리스트 $\{I_0, I_1, \cdots, I_{n-1}\}$에 대해서, 이동 객체 A와 B사이의 관계 궤적, $RT^{(A,B)}$는 다음과 같이 표현된다.

$$\begin{aligned} RT^{(A,B)} &= \{P_0^{(A,B)}, P_1^{(A,B)}, \ldots, P_n^{(A,B)}\} + \{M_0, M_1, \ldots, M_{n-1}\} \\ &= \{(L_0^{(A)}, A_0^{(A)}, L_0^{(B)}, A_0^{(B)}, R_0, \alpha_0), (L_1^{(A)}, A_1^{(A)}, L_1^{(B)}, A_1^{(B)}, R_1, \alpha_1), \\ &\quad \cdots, (L_n^{(A)}, A_n^{(A)}, L_n^{(B)}, A_n^{(B)}, R_n, \alpha_n)\} + \{(D_0, I_0), (D_1, I_1), \cdots, \\ &\quad (D_{n-1}, I_{n-1})\} \end{aligned}$$

[정의 1]과 [정의 2]를 기반으로 둘 이상의 객체로 이루어진 다수개의 궤적인 다중 궤적, MT는 다음과 같이 정의한다.

[정의 3] 객체 A_1, 객체 A_2, $\cdots$, 객체 A_n 중에 적어도 하나는 이동 객체라고 할 때, 이동 객체 $A_1, A_2, \cdots, A_n$으로 이루어진 다중 궤적, $MT^{(A1, A2, \cdots, An)}$는 다음과 같이 정의한다.

$$MT^{(A1,\ A2,\ \ldots,\ An)} = \sum_{i=1}^{n} ST^{(A_i)} + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} RT^{(A_i, A_j)}$$

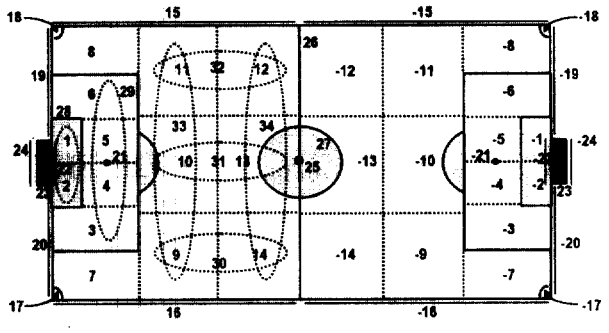
여기서, $ST^{(A1)}$, $ST^{(A2)}$, $\cdots$, $ST^{(An)}$는 각 객체의 단일 궤적을 나타내며, $RT^{(A1,A2)}$, $RT^{(A1,A3)}$, $\cdots$, $RT^{(An-1,An)}$는 각각 객체들간의 관계 궤적을 나타낸다.

4. 내용 및 개념 기반 검색

본 논문에서는 제안한 시공간 표현 기법의 유용성을 보이기 위해, 축구 비디오 데이터베이스를 응용 분야로 선정한다. 축구 비디오 데이터는 사용자의 주된 관심 객체인 축구공이 경기장을 배경으로 많은 움직임 정보를 지니고 있기 때문에, 객체의 궤적을 추출하는 데 있어 매우 용이할 뿐만 아니라, 축구공의 궤적이 발생하는 위치에 따라 축구 경기에서 일어나는 개념(의미) 정보를 추출할 수 있다. 이를 기반으로 본 논문에서 지원하는 사용자 검색 질의를 크게 내용 기반 검색과 개념 기반 검색으로 나누어 설명한다.

4.1 내용 기반 검색

효율적인 내용 및 개념 기반 검색을 지원하기 위해, 먼저 (그림 4)와 같이 축구 경기장의 내용 및 개념 기반 위치 정보를 정의한다. (그림 4)의 (a)는 사용자의 주된 관심 객체인 축구공이나 주요 선수들의 궤적을 모델링하는 데 있어 중요한 요소인 위치 정보를 보다 효율적으로 표현하고, 아울러 축구 비디오 데이터에서 일어나는 골킥, 코너킥, 드로잉과 같은 주요 개념(의미)을 효과적으로 추출하기 위해서, 본 논문에서 설계한 축구 경기장 모델을 나타내며, 실제로 이는 응용에 따라 변경이 가능하다. (그림 4)의 (b)는 (그림 4)의 (a)의 축구 경기장 모델에서의 각각의 위치에 해당하는 이름을 나타낸다.



(a) 축구 비디오 데이터를 위한 축구 경기장 모델

1	Left Upper Goal	LUG(RUG)
2	Left Lower Goal	LLG(RLG)
3	Left Bottom Penalty	LBP(RBP)
4	Left Lower Penalty	LLP(RLP)
5	Left Upper Penalty	LUP(RUP)
6	Left Top Penalty	LTP(RTP)
7	Left Left Bottom	LLB(RRB)
8	Left Left Top	LLT(RRT)
9	Left Center Bottom	LCB(RCB)
10	Left Center Middle	LCM(RCM)
11	Left Center Top	LCT(RCT)
12	Left Right Top	LRT(RLT)
13	Left Right Middle	LRM(RLM)
14	Left Right Bottom	LRB(RLB)
15	Left Top Sideline	LTS(RTS)
16	Left Bottom Sideline	LBS(RBS)
17	Left Bottom Corner	LBC(RBC)

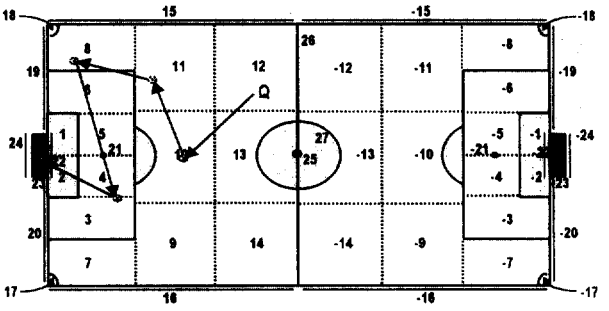
18	Left Top Corner	LTC(RTC)
19	Left Top Goalline	LTG(RTG)
20	Left Bottom Goalline	LBG(RBG)
21	Left Penalty Point	LPP(RPP)
22	Left Goalpost Line	LGL(RGL)
23	Left Goalpost Inside	LGI(RGI)
24	Left Goalpost Over	LGO(RGO)
25	Center Circle Point	CCP
26	Half Line	HFL
27	Center Circle Area	CCA
28	Left Goal Area	LGA(RGA)
29	Left Penalty Area	LPA(RPA)
30	Left Bottom Area	LBA(RBA)
31	Left Middle Area	LBA(RBA)
32	Left Top Area	LTA(RTA)
33	Left Left Area	LLA(RRA)
34	Left Right Area	LRA(RLA)

(b) 내용 및 개념 기반 검색을 위한 위치 정보

(그림 4) 내용 및 개념 기반 검색을 위한 축구 경기장의 위치 정보

본 논문에서 지원하는 내용 기반 검색 질의는 축구 비디오 데이터에서 추출한 이동 객체의 궤적을 기반으로 하는 검색 질의를 의미하며, 움직임 정보를 가지는 비디오 데이터의 특성상 매우 중요한 형태의 사용자 질의에 속하며, 사용자는 사용자 인터페이스(GUI) 상에서 사용자가 원하는 형태의 궤적을 표현함으로써 질의를 표현할 수 있다. 즉, "*사용자 인터페이스를 통해 사용자가 스케치한 궤적과 유사한 궤적을 가진 모든 축구 비디오 샷을 찾아라.*"의 질의가 이에 해당한다. 예를 들어, 사용자 인터페이스를 통해 (그림 5)와 같은 질의 궤적 Q가 주어지면, 질의 궤적 Q를 본 논문에서 제안하는 시공간 표현 기법을 이용하여 다음과 같이 표현한 후, 이를 기반으로 축구 비디오 데이터베이스를 탐색해서 유사한 궤적을 가지는 모든 비디오 샷을 검색한다.

$$ST^{(Q)} = \{P_n^{(Q)}\} + \{M_{n-1}\}$$
$$= (L_0^{(Q)}, \text{객체 A}, L_1^{(Q)}, \text{객체 A}, L_2^{(Q)}, \text{객체 A}, L_3^{(Q)}, \text{객체 A}, L_4^{(Q)}, \text{객체 A}, L_5^{(Q)}) + (M_0, M_1, M_2, M_3, M_4)$$
$$= (\text{LRT}, \text{객체 A}, \text{LCM}, \text{객체 A}, \text{LCT}, \text{객체 A}, \text{LLT}, \text{객체 A}, \text{LLP}, \text{객체 A}, \text{LGI}) + \{(240, 19, I_0), (105, 19, I_1), (150, 12, I_2), (280, 32, I_3), (135, 16, I_4)\}$$



(그림 5) 궤적을 이용한 내용 기반 검색 질의 예

4.2 개념 기반 검색

축구 비디오 데이터베이스에서는 일련의 축구공의 궤적과 4.1절에서 보여진 바와 같이 궤적이 발생하는 축구 경기장 모델의 위치 정보를 토대로 코너킥(Corner Kick), 패널티 킥(Penalty Kick), 골킥(Goal Kick), 드로우 인(Throw In), 프리 킥(Free Kick), 센터링(Centering) 등의 개념(의미)을 유추할 수 있다. 따라서, 본 논문에서는 궤적을 이용한 내용 기반 검색뿐 아니라, 일반적으로 축구 비디오 데이터에서 중요하게 다루는 개념들을 이용한 개념 기반 검색을 지원함으로써 검색의 정확성과 더불어 사용자의 편리성을 도모할 수 있다. 즉, "*프리킥이나 혹은 슛팅을 하는 장면을 포함하고 있는 모든 축구 비디오 샷을 찾아라.*"의 질의가 이에 해당한다. 따라서, 사용자가 위와 같은 개념 기반 검색을 원하면, 이를 처리하는 시스템은 주어진 질의 개념을 해당하는 시공간 표현 기법으로 변환한 후, 4.1절에서 언급한 내용 기반 검색 질의를 처리하는 것과 같은 과정을 거쳐 검색을 수행한다. 다음 <표 2>

는 축구 비디오에서 다루는 주요 개념과 그에 해당하는 시공간 표현 기법을 나타낸다. 단, 경기장 왼쪽 진영을 가정하며 오른쪽 진형도 이와 유사하다. 여기서, '&', '|', '?', '[]' 연산자는 각각 And, Or, Don't care, All을 의미한다.

또한, 본 논문에서는 축구 비디오 데이터의 경우 주요 이동 객체인 축구공을 소유하고 있는 주요 선수를 기반으로 검색 질의가 가능한 행위자(Actor) 기반 검색 질의를 지원한다. 즉, "*선수 A가 축구공을 가지고 있는 장면을 포함하고 있는 모든 축구 비디오 샷을 찾아라.*" 또는, "*선수 B가 어시스트한 후, 슛팅을 해서 골인되는 장면을 포함하고 있는 모든 축구 비디오 샷을 찾아라.*"의 질의가 이에 해당한다.

〈표 2〉 축구 경기에서의 주요 개념과 그것의 시공간 표현

개 념	시공간 표현
골 킥 (Goal Kick)	(LGA|LPA, [!(LGA&LPA)]) + {(0~90|270~360, ?, I_0)}
페널티 킥 (Panelty Kick)	a. (LPP, LPG|LBG|LGO|LGI) + {(90~270, ?, I_0)} b. (LPP, LPL, LGA|LPA) + {(90~270, ?, I_0), (0~90|270~360, ?, I_1)}
프리 킥 (Free Kick)	(LLB|LLT|LBA|LMA|LTA, LGA|LPA) + {(90~270, ?, I_0)}
센터링 (Centering)	(LLB|LLT|LCB|LCT, LGA|LPA) + {(90~270, ?, I_0)}
코너 킥 (Corner Kick)	a. (LTC,LLT|LGA|LPA|LTG|LGL|LGI) + {(270~360, ?, I_0)} b. (LBC,LLB|LGA|LPA|LBG|LGL|LGI) + {(0~90, ?, I_0)}
드로우 인 (Throw In)	a. (LBS, [!LBS]) + {(0~180, ?, I_0)} b. (LTS, [!LTS]) + {(180~360), ?, I_0)}
골 인 (Goal In)	(?, ···, LGI) + {(?, ?, I_0), ···, (?, ?, I_{n-1})}
골 아웃 (Goal Out)	(?, ···, LTG|LBG) + {(?, ?, I_0), ···, (?, ?, I_{n-1})}

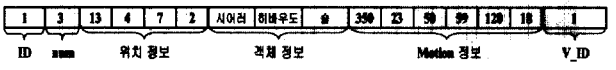
5. 이동 객체를 위한 시그니처 기반 접근 기법(Signature-based Access method for Moving Object's Trajectory, SAMOT)

데이터베이스가 점점 대용량화 되어감에 따라, 비디오 데이타와 같은 복합적인 구조를 가지는 멀티미디어 데이터에 대한 주된 연구는 주어진 사용자 질의에 대해서 검색의 효율성을 어떻게 보장할 수 있는 가이다. 이를 위해, 본 논문에서는 기존의 대규모 데이터베이스에서 많이 사용되었던 시그니처 방식을 이용한 이동 객체를 위한 새로운 시그니처 기반 접근 기법(Signature-based Access method for Moving Object's Trajectory, SAMOT)을 제안한다. 제안하는 SAMOT 접근 기법은 데이터 파일에서 움직임 정보들에 대한 순차 탐색을 수행하기 전에, 각각의 움직임 정보에 대한 시그니처를 탐색하여 필터링을 수행한 후, 검색된 후보 시그니처들만을 이용하여 디스크를 접근하기 때문에 데이터 파일에 대한 디스크 접근 횟수를 최소화시킴으로 인해 빠른 검색 성능을 제공함은 물론 이동 객체를 이용한 사용자의 다양한 질의를 효율적으로 처리할 수 있도록 설계한다.

5.1 SAMOT 접근 기법의 구조

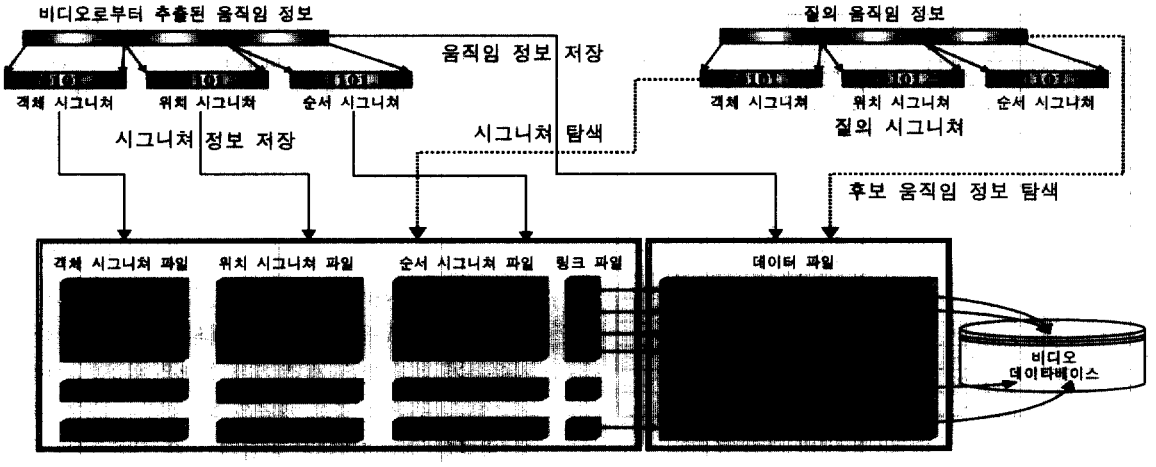
SAMOT 접근 기법은 크게 두 부분으로 나뉜다. 즉, 비디오 샷으로부터 추출된 이동 객체의 움직임 정보들을 저장하고 있는 데이터 파일(*.mot) 부분과 움직임 정보에 대한 시그니처를 생성하여 저장하는 시그니처 파일(*.sig) 부분이다. 먼저, 데이터 파일의 자료 저장 구조는 다음 (그림 6)과 같다.

```
struct _mot {
  int ID ;                                  // 식별자
  int num ;                                 // 동작의 수
  int locList[MAX_LOC] ;                    // 동작이 발생한 위치 정보
  char objList[MAX_OBJ][MAXNAME] ;          // 위치 정보에서의 객체 정보
  Motion motionList[MAX_MOTION] ;           // 동작(Motion) 정보
  RecID V_ID ;                              // 실제 비디오 샷을 가리키는 식별자
} T_MOT ;
```



(그림 6) 움직임 정보를 저장하는 데이터의 자료 저장 구조의 예

여기서, ID는 비디오 샷으로부터 추출된 움직임 정보를 저



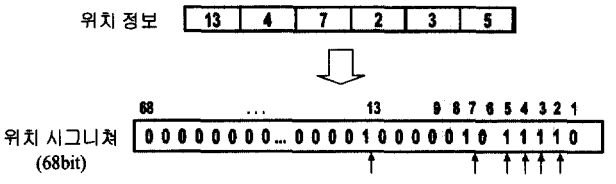
(그림 7) SAMOT 접근 기법의 전체적인 구조

장하는 데이터 파일에서의 레코드 식별자를 의미하고, num은 움직임 정보를 이루는 동작(Motion)의 수를 나타낸다. 위치정보는 이동 객체의 위치 정보 즉, 본 논문에서 제안하는 축구 비디오 데이터를 위한 운동장 모델에서의 위치 정보를 의미한다. 객체 정보는 각각의 위치 정보에서 이동 객체를 소유하고 있는 행위자(Actor) 즉, 축구 비디오 데이터베이스의 경우 축구공을 가지고 있는 선수를 의미한다. 동작(Motion) 정보는 이동 객체가 움직인 방향(Real Angle)과 거리(Distance)의 쌍으로 구성된다. 마지막으로 V_ID는 해당 움직임 정보가 추출된 실제 축구 비디오 데이터 스트림을 가리키는 객체 식별자를 의미한다.

움직임 정보를 위한 시그니쳐 파일 부분은 세부적으로 4개의 파일로 구성된다. 첫째, 해당 이동 객체의 움직임 정보들 중에서 객체 정보에 해당하는 시그니쳐를 생성해서 저장하는 객체 시그니쳐 파일(*_obj.sig) 둘째, 위치 정보에 해당하는 시그니쳐를 생성해서 저장하는 위치 시그니쳐 파일(*_loc.sig) 셋째, 위치 정보에 순서 정보(Order Information)까지 포함해서 시그니쳐를 생성해서 저장하는 순서 시그니쳐 파일(*_ord.sig) 마지막으로, 시그니쳐 파일과 데이터 파일을 연결하기 위한 링크 파일(*.lnk)이다. 본 논문에서 제안하는 SAMOT 접근 기법의 전체적인 구조는 (그림 7)과 같다.

5.2 시그니쳐 생성

본 논문에서는 이동 객체의 움직임 정보에 대해서 각각 객체 시그니쳐(SIG_{obj}), 위치 시그니쳐(SIG_{loc}), 그리고 순서 시그니쳐(SIG_{ord})로 나누어 시그니쳐를 생성한 후, 각각의 시그니쳐 파일에 따로 분리해서 저장한다. 이렇게 함으로써 사용자 질의에 따라 이에 부합하는 해당 시그니쳐 파일만을 탐색함으로써 전체적인 검색 성능을 향상시킨다. 첫째, 움직임 정보들 중에서 객체 정보에 기반해서 생성한 객체 시그니쳐는 움직임 정보를 이용한 사용자의 질의 중에서 객체 정보를 이용한 질의를 효율적으로 처리하기 위해서이다. 둘째, 위치 시그니쳐는 이동 객체의 움직임 정보들 중에서 움직임이 발생한 위치를 고려하여 생성한다. 위치 시그니쳐를 생성하는 방법은 (그림 4)의 (a)에서와 같이 축구 경기장 모델을 기반으로 축구공의 움직임이 발생한 위치에 해당하는 번호에 따라 해당 비트를 1로 세팅시키는 방법이다. (그림 8)은 위치 시그니쳐를 생성하는 예를 보여준다. 마지막으로, 순서 시그니쳐는 이동 객체의 움직임 정보들 중에서 움직임이 발생한 위치와 더불어 순서까지 함께 고려하여 생성한다. 순서 시그니쳐를 생성하는 방법은 먼저, (그림 4)의 (a)에서와 같이 축구 운동장 모델에서의 위치 정보 68가지에 대해서 각각 하나의 아스키 문자로 매핑시킨 후, 순서 정보를 같이 결합해서 이를 문자열로 변환한 것을 기반으로 순서 시그니쳐를 생성한다. 예를 들어 축구 비디오 샷으로부터 추출된 축구공의 움직임 위치가 다음과 같이 "4 10 2 5 3"으로 이루어져 있다면, 각각 위치에 맞는 해당 아스키 문자를 다음과 같이 'D', 'J', 'B', 'E', 'C' 생성한다. 그런 후에, 여기에 순서 정보를 같이 결합해서 다음과 같이 "D1", "J2", "B3", "E4", "C5"과 같은 최종적인 순서 정보 리스트를 생성하고, 이를 토대로 순서 시그니쳐를 생성한다. 객체 시그니쳐, 위치 시그니쳐, 그리고 순서 시그니쳐를 생성하는 알고리즘은 다음과 같다.



(그림 8) 위치 시그니쳐 생성 예

```
int genSignature(inMotDat, sigObj, sigLoc, sigOrd)
T_MOT inMotData   // 움직임 정보를 담은 구조체
unsigned char *sigObj ;   // 생성해서 반환할 객체 시그니쳐
unsigned char *sigLoc ;   // 생성해서 반환할 위치 시그니쳐
unsigned char *sigOrd ;   // 생성해서 반환할 순서 시그니쳐
{
  int k ;
  unsigned char *tSigObj, *tSigLoc, *tSigOrd ;  // 시그니쳐를 저
                                                장할 임시 변수

  tSigObj, tSigLoc 그리고 tSigOrd를 위한 메모리 할당 ;
  for(k=0 ; k<n ; k++) {
    // sigObjBitSize, sigLocBitSize, sigOrdBitSize : 객체, 위치, 순
       서 시그니쳐를 위한 시그니쳐 비트 수
    // 초기화시키는 함수 호출
    clearSig(tSigObj, sigObjBitSize) ;
    clearSig(tSigLoc, sigLocBitSize) ;
    clearSig(tSigOrd, sigOrdBitSize) ;
    // sigObjSetBitSize, sigLocSetBitSize, sigOrdSetBitSize :
       전체 시그니쳐 비트 수 중에서 1로 세팅할 비트수
    // k번째 객체, 위치, 순서에 대한 각각의 객체, 위치, 순서 시그
       니쳐를 생성하는 함수 호출
    genKthObjSignature(tSigObj, inMotData.objList[k], sigObj-
    BitSize, sigObjSetBitSize) ;
    genKthLocSignature(tSigLoc, inMotData.locList[k], sigLoc-
    BitSize, sigLocSetBitSize) ;
    genKthOrdSignature(tSigOrd, inMotData.ordList[k], sigOrd-
    BitSize, sigOrdSetBitSize) ;
    // 각각의 생성된 k번째 시그니쳐를 시그니쳐 비트 수만큼 Or
       연산을 수행하는 함수 호출
    sigOring(sigObj, tSigObj, sigObjBitSize) ;
    sigOring(sigLoc, tSigLoc, sigLocBitSize) ;
    sigOring(sigOrd, tSigOrd, sigOrdBitSize) ;
  }
  tSigObj, tSigLoc 그리고 tSigOrd를 위해 할당된 메모리 해제 ;

  return SUCCESS ;
}
```

(Algorithm 1) Signature Generation Algorithm

5.3 삽입 및 검색 알고리즘

주어진 새로운 이동 객체의 움직임 정보들을 SAMOT 접근 기법에 삽입하기 위해서는 (그림 7)에서와 같이 먼저 움직임 정보들을 기반으로 앞에서 언급한 시그니쳐 생성 알고리즘을 이용해 객체 시그니쳐, 위치 시그니쳐 그리고 순서 시

그니쳐를 생성한 후, 생성된 시그니쳐들을 각각의 해당 시그니쳐 파일에 삽입한다. 그런 다음, 실제 움직임 객체의 움직임 정보를 데이터 파일에 삽입한 후, 검색 시에 시그니쳐 파일을 통해 검색된 후보 시그니쳐들을 통해서 실제 데이터 파일에 있는 해당 움직임 정보들을 접근하기 위해 필요한 링크 파일에 데이터 파일에서의 해당 움직임 정보의 저장 위치와 크기 정보를 저장하는 것으로 삽입 과정을 마친다. 삽입 과정을 알고리즘으로 표현하면 다음과 같다.

```
int insert(inMotData)
T_MOT inMotData ;          // 움직임 정보를 담은 구조체
{
  FILE *objSigFp, *locSigFp, *ordSigFp ; // 객체, 위치, 순서 시그
                                          니쳐 파일 포인터
  FILE *motFp, *lnkFp ;                  // 데이터 파일, 링크 파
                                          일 포인터
  unsigned char *sigObj, *sigLoc, *sigOrd ; // 객체,위치,순서시그니쳐

  unsigned long Len ;                    // 움직임 정보의 크기
  unsigned long Offset ;                 // 데이터 파일에서의 저
                                          장 위치

  // 객체 시그니쳐를 생성한 후, 객체 시그니쳐 파일에 삽입하기 위한
     함수 호출
  if(genObjSignature(objSigFp, inMotData.objList, inMotData.num,
  sigObj) == FAIL)
      return FAIL ;
  // 위치 시그니쳐를 생성한 후, 위치 시그니쳐 파일에 삽입하기 위한
     함수 호출
  if(genLocSignature(locSigFp, inMotData.locList, inMotData.num,
  sigLoc) == FAIL)
      return FAIL ;
  // 순서 시그니쳐를 생성한 후, 순서 시그니쳐 파일에 삽입하기 위한
     함수 호출
  if(genOrdSignature(ordSigFp, inMotData.locList, inMotData.num,
  sigOrd) == FAIL)
      return FAIL ;
  // 움직임 정보를 데이터 파일에 저장
  if(storeMotData(motFp, inMotData, &Offset, &Len) == FAIL)
      return FAIL ;
  // 링크 파일에 움직임 정보의 크기(Len)와 데이터 파일에서의 파일
     위치(Offset)를 저장
  if(storeLink(lnkFp, Offset, Len) == FAIL)
      return FAIL ;

  return SUCCESS ;
}
```

(Algorithm 2) Insertion Algorithm

이동 객체의 움직임 정보를 이용한 사용자의 질의 처리는 다음과 같다. 먼저, 질의 움직임 정보가 주어지면, 질의 움직임 정보를 이용해서 질의 시그니쳐 즉, 질의 객체 시그니쳐, 질의 위치 시그니쳐, 그리고 질의 순서 시그니쳐를 생성한다. 이렇게 생성된 질의 시그니쳐들을 이용하여 시그니쳐 파일에 저장된 각각의 시그니쳐들을 순차적으로 탐색하여 필터링을 수행한다. 그러나, 사용자 질의 타입에 따라 질의를 처리하기 위해 필요한 시그니쳐 파일만을 탐색하게 된다. 물론, 모든 시그니쳐 파일을 모두 탐색해야 할 경우는 먼저 객체 시그니쳐 파일로부터 후보 시그니쳐들만을 선택한다. 그리고 선택된 후보들만을 이용해서 두 번째로 위치 시그니쳐 파일을 탐색한다. 그리고 마지막으로는 두 단계의 시그니쳐 파일 탐색을 거쳐서 얻은 후보들을 이용해서 순서 시그니쳐 파일을 탐색하게 된다. 따라서, 이렇게 시그니쳐 파일 탐색을 통해 최종적으로 선택된 후보 시그니쳐들만을 가지고 데이터 파일을 접근해서 해당하는 실제 움직임 정보들을 탐색한다. 그런 후에, 시그니쳐 기법의 특성상 검색 매치 오류(False Drop)[15]를 감안해서 검색 매치 오류를 체크한 후, 최종적으로 주어진 사용자 질의에 만족하는 움직임 정보들만을 이용해서 사용자 질의와 검색된 최종 결과와의 유사성을 계산한다. 그리고 유사성이 높은 순으로 정렬해서 검색 결과를 반환한다. 검색 과정을 알고리즘으로 표현하면 다음과 같다.

```
int retrieval(qryMotData, result)
T_MOT qryMotData ; // 질의 움직임 정보를 담은 구조체
T_RES result[] ;       // (id, v_id, sim_weight)를 포함한 검색 결과
{
  // 질의 객체 시그니쳐, 질의 위치 시그니쳐, 질의 순서 시그니쳐 저
     장할 변수
  unsigned char *qrySigObj, *qrySigLoc, *qrySigOrd ;
  // 후보 시그니쳐를 통해 데이터 파일로부터 읽어온 움직임 정보를
     담을 배열
  T_MOT dbMotList[] ;
  int n ;                                  // 검색된 후보 결과의 수
  int k ;

  // 주어진 질의 움직임 정보에 대한 각각의 질의 객체 시그니쳐, 질
     의 위치 시그니쳐,
  // 질의 순서 시그니쳐를 생성
  if(genQuerySignature(qryMotData, qrySigObj, qrySigLoc, qry-
  SigOrd) == FAIL)
     return FAIL ;
  // 질의 객체 시그니쳐를 이용해 객체 시그니쳐 파일을 탐색한 후,
     후보 결과를 반환 (1차 필터링)
  if(searchObjSignature(qrySigObj, result, &n) == FAIL)
     return FAIL ;
  // 앞서 얻은 후보 결과를 토대로 질의 위치 시그니쳐를 이용해 위
     치 시그니쳐 파일을
  // 탐색한 후, 후보 결과를 반환(2차 필터링)
  if(searchLocSignature(qrySigLoc, result, &n) == FAIL)
     return FAIL ;
  // 앞서 얻은 후보 결과를 토대로 질의 순서 시그니쳐를 이용해 순
     서 시그니쳐 파일을
  // 탐색한 후, 후보 결과를 반환(3차 필터링)
  if(searchOrdSignature(qrySigOrd, result, &n) == FAIL)
     return FAIL ;
  // 시그니쳐를 통해 필터링된 검색 후보 결과를 이용해 데이터 파일
     로부터 움직임 정보를 읽음
  if(readMotData(result, dbMotList, n) == FAIL)
     return FAIL ;
  for(k = 0 ; k < n ; k++) {
     // 시그니쳐의 특성상 검색 매치 오류(False Drop)를 체크
     if(isNotFalseDrop(dbMotList[k], qryMotData) == SUCCESS) {
        if(calcSimilarity(dbMotList[k], qryMotData, result[k]) ==
        FAIL)
           return FAIL ;
     }
  }
  // 검색된 결과에 대해서 유사성에 준하여 정렬
  if(sortResult(result, n) == FAIL)
     return FAIL ;

  return SUCCESS ;
}
```

(Algorithm 3) Retrieval Algorithm

6. 실험 및 성능 평가

본 장에서는 제안하는 SAMOT 접근 기법의 검색 성능의 효율성을 검증하기 위해 실험을 수행한 실험 환경 및 성능 평가를 위해 사용하는 실험 데이터 셋, 그리고 성능 평가 지수에 대해서 기술한다. 특히, 성능 평가 지수는 크게 두 가지 측면 즉, 검색 효과(Retrieval Effectiveness)와 검색 효율(Retrieval Efficiency)측면[16]을 고려한다. 검색 효과 측면에서는 평균 정확율(Average Precision)과 평균 재현율(Average Recall)을 고려하며, 검색 효율 측면에서는 삽입 시간(Insertion Time), 검색 시간(Retrieval Time), 디스크 접근 횟수(Disk Access I/O), 그리고 부가 저장 공간(Storage Overhead)을 고려한다.

6.1 실험 환경 및 데이터 셋

본 논문에서는 128MB 메인 메모리를 탑재한 펜티엄 Ⅲ-800의 Windows 2000 운영체제 하에서 실험한다. 아울러, 성능 평가를 위한 실험 데이터 셋은 크게 세 가지 즉, 소규모 리얼(Small Real) 데이터 셋, 대규모 리얼(Large Real) 데이터 셋 그리고 Synthetic 데이터 셋으로 분류한다. 먼저, 소규모 리얼 데이터 셋은 mpeg 동영상 파일 형식으로 대부분이 "골인되는(Goal In) 장면"을 포함하고 있는 실제 축구 비디오 데이터 350개로 구성되어 있으며, 각각의 축구 비디오 데이터에는 1개에서 15개의 움직임 동작(Motions)들로 구성되어 있다. 이 데이터 셋은 검색 효과 즉, 정확율과 재현율 측면에서 성능 평가를 측정하는 데 사용한다. 그리고 대규모 리얼 데이터 셋은 앞서 설명한 소규모 리얼 데이터 셋 350개를 확장시켜 생성한 460,000건의 데이터 셋이다. 마지막으로, Synthetic 데이터 셋은 임의의 데이터를 생성할 수 있는 무작위 데이터 생성 프로그램을 통해서 얻은 균등 분포 특성을 가진 임의의 데이터 셋 100,000건이다. 검색 효과를 측정하기 위해서 소규모 리얼 데이터 셋을 사용하며, 이는 데이터 셋이 클 경우, 질의를 만족하는 관련 비디오 샷(Relevant Data Set)을 수작업으로 추출할 수 없기 때문이다. 아울러, 검색 효율을 측정하기 위해 대규모 리얼 데이터 셋과 Synthetic 데이터 셋을 사용하며, 이는 데이터의 수가 적은 경우(10만 이하) 삽입 시간, 검색 시간, 부가 저장 공간 측면에서의 성능 평가가 의미가 없기 때문이다. <표 3>은 본 논문에서 사용하는 실험 인자를 나타낸다

〈표 3〉 실험에 사용되는 실험 인자

인자 \ 데이터 셋 종류	소규모 리얼 데이터 셋	대규모 리얼 데이터 셋	Synthetic 데이터 셋
실험 데이타의 수	350건	460,000건	100,000건
검색 종류	순차 검색	순차 검색(SeqScan)과 SAMOT 접근 기법을 이용한 검색	
성능 평가 지수	검색 효과	검색 효율	
움직임의 수	1개 ~ 15개		
질의의 수	40개	100개	100개
객체 시그니쳐 비트수	N/A	61 bit	59 bit
움직임 시그니쳐 비트수	N/A	68 bit	
순서 시그니쳐 비트수	N/A	37 bit	
디스크 페이지 크기	4096B(4KB)		

6.2 성능 평가 : 검색 효과(Retrieval Effectiveness)

본 논문에서는 검색 효과를 평가하기 위하여 정확율과 재현율을 사용한다. 그리고 타 연구인 Li 방법과 Shan 방법을 제안하는 방법과 비교 수행한다. 이를 위해, 먼저, RVQ(Relevnat Video data to Query)는 주어진 질의에 대한 시스템의 검색한 검색 결과의 수, RVD(Relevnat Video data in Database)는 주어진 질의에 대해 유사성이 있어 데이터베이스로부터 반드시 검색되어져야 할 검색 결과의 수, 마지막으로 RVR(Relevnat Video data that are Retrieved)는 질의에 대해 시스템이 검색한 결과 중에서 유사성을 가지고 있는 검색 결과의 수 즉, RVQ 중에서 RVD에 속하는 결과의 수라고 정의한다. 아울러, RVD를 구하기 위해 10명의 컴퓨터 관련 대학원생을 통하여 수작업를 통해 주어진 질의에 대해 축구 비디오 데이터베이스로부터 유사성이 높은 축구 비디오 데이터를 선택하도록 한다. 따라서, 다음 식 (1)은 정확율과 재현율을 계산하는 식이다.

$$\text{정확율}(Precision) = \frac{RVR}{RVQ} \quad \text{재현율}(Recall) = \frac{RVR}{RVD} \quad (1)$$

본 논문에서는 보다 정확한 정확율과 재현율을 구하기 위하여 가장 널리 사용되는 평균 11 point 방법[17]을 사용하였다. <표 4>는 각각의 방법에 대한 평균 정확율과 평균 재현율을 구한 결과이다. 실험 결과를 통해 알 수 있듯이 제안하는 방법이 기존의 Li 방법보다 정확율과 재현율 모든 측면에서 우수함을 알 수 있다. 다시 말하면, 제안하는 방법이 정확율 측면에서는 약 20%, 재현율 측면에서는 약 10% 정도의 검색 효율이 좋아짐을 볼 수 있다. 또한, Shan 방법보다는 재현율 측면에서는 비슷하나 정확율 측면에서는 약 17% 정도의 더 나은 검색 성능을 보인다.

〈표 4〉 검색 효과 비교

	평균 정확율	평균 재현율
Li 방법	23 %	42 %
Shan 방법	26 %	46 %
제안하는 방법	43 %	44 %

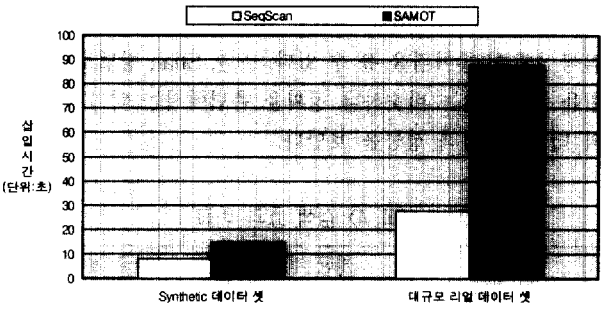
6.3 성능 평가 : 검색 효율(Retrieval Efficiency)

검색 효율 측면에서의 성능 평가는 삽입 시간, 검색 시간, 그리고 부가 저장 공간의 3가지로 나누어서 수행한다. Li 방

법이나 Shan방법과 같이 타 연구에서는 빠른 검색 성능을 지닌 접근 기법을 사용하지 않기 때문에 일반적으로 순차 접근 방식(Sequential Scan Method : SeqScan)을 이용해서 검색하는 것으로 간주한다.

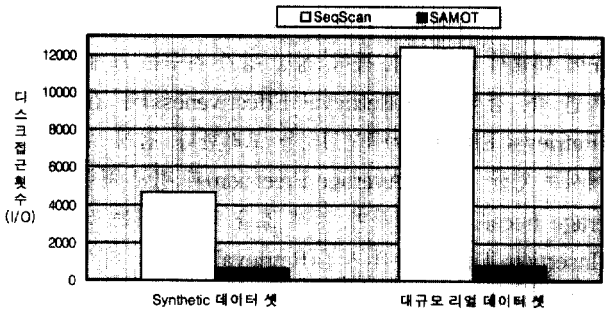
먼저, 삽입 시간은 실제 축구 비디오 샷으로부터 추출된 움직임 정보들 즉, 움직임 발생한 위치 정보(Location), 그 위치에서 움직임 객체를 소유한 객체명(예> 축구공을 소유한 선수이름), 움직임 객체의 움직임 방향과 거리로 구성된 움직임 정보로 구성된 데이터를 시그니쳐 파일과 데이터 파일에 각각 저장하는 데 소요되는 시간을 의미한다. (그림 9)는 각각 Synthetic 데이터 셋과 대규모 리얼 데이터 셋을 삽입하는 데 소요되는 시간을 나타낸다. SeqScan 방식은 접근 기법을 사용하지 않는 순차적인 저장 방식이기 때문에 데이터 파일에만 저장하는 시간을 의미한다. 그에 반해 SAMOT 접근 기법은 시그니쳐 파일과 데이터 파일에 모두 저장하는 데 걸리는 시간이다. 여기서, 세로축은 삽입 시간을 초 단위로 나타낸 것이며, 삽입 시간은 메모리에서의 연산 시간(CPU 시간)과 디스크 접근 시간을 모두 포함한 전체 시간(Wall 시간)을 의미한다. 대규모 리얼 데이터 셋의 경우 SeqScan 방식이 삽입 시간은 약 27초, SAMOT 접근 기법은 약 87초 정도 소요된다. 이는 움직임 정보들을 데이터 파일에 저장하기 전에 먼저, 시그니쳐를 구성한 후, 구성된 시그니쳐들을 각각의 시그니쳐 파일에 저장하는 시간을 요구하기 때문이다.

둘째, 검색 성능을 평가하기 위해 Synthetic 데이터 셋과 대규모 리얼 데이터 셋으로부터 무작위로 각각 100개의 질의 움직임 정보를 추출한 것을 이용해 평균값을 구해서 검색에 요구되는 디스크 접근 횟수(Disk Access I/O)와 탐색 시간(Search Time)으로 나누어 성능을 측정한다. 여기서 SeqScan 방식의 경우는 움직임 정보들로 구성된 데이터 파일만을 탐색하며, 한번 디스크를 읽어오는 경우 디스크 페이지 크기인 4096B(4KB)로 가정한다.



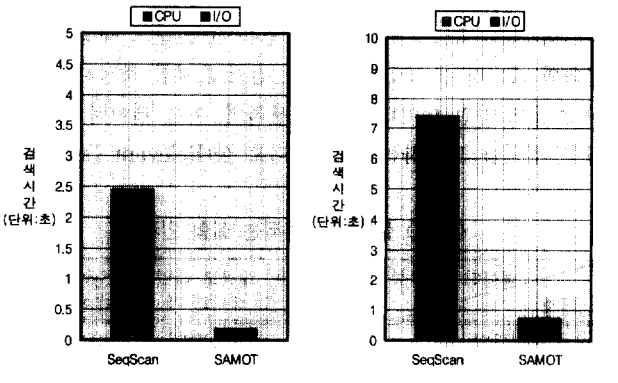
(그림 9) 삽입 시간

(그림 10)은 Synthetic 데이터 셋과 대규모 리얼 데이터 셋에 대해서 각각 100개 질의에 대해서 평균을 구한 디스크 접근 횟수를 나타낸다. 두 데이터 셋 모두에 대해서 SAMOT 접근 기법이 SeqScan 방식에 비해 월등히 좋은 성능을 보인다. 특히 SAMOT 접근 기법이 데이터의 수가 많은 대규모 리얼 데이터 셋의 경우 SeqScan 방식보다 훨씬 적은 수의 디스크 접근 횟수를 요구한다.



(그림 10) 질의 검색에 대한 디스크 접근 횟수(I/O)

(그림 11)은 검색 질의에 대한 전체 응답 시간을 메모리에서 계산하는 데 소요되는 CPU 시간과 데이터 파일에서 움직임 정보를 읽어오는 데 걸리는 디스크 접근 I/O 시간으로 나누어 검색 시간을 비교 평가한다. (그림 11)의 (a)는 Synthetic 데이터 셋에 대한 검색 시간을 측정한 결과로 SeqScan 방식의 경우 CPU 시간과 I/O 시간 모두 약 1.2초 걸리는 데 반해 SAMOT 접근 기법의 경우 CPU 시간은 약 0.2초, I/O 시간은 0.05초 정도 밖에 소요되지 않는다. 전체 시간을 고려해 볼 경우 SAMOT 접근 기법이 SeqScan 방식에 비해 약 10배 정도의 검색 성능을 보인다. (그림 11)의 (b)는 대규모 리얼 데이터 셋에 대한 검색 시간을 측정한 결과로 Synthetic 데이터 셋과 거의 비슷한 결과를 나타낸다. 아울러, 데이터 수의 크기에 따라 SeqScan 방식은 검색 시간의 차이가 큰 반면에 SAMOT 접근 기법의 경우 거의 유사한 성능을 보인다.



(a) Synthetic 데이터 셋 (b) 대규모 리얼 데이터 셋

(그림 11) 검색 질의에 대한 응답 시간(CPU 시간 + I/O 시간)

〈표 5〉 SAMOT을 통해 검색된 후보 결과

데이터 셋 \ 시그니쳐	위치 시그니쳐	위치 + 순서 시그니쳐	최종 결과
Synthetic 데이터 셋	2,181	638	627
대규모 리얼 데이터 셋	10,358	800	796

<표 5>는 SAMOT 접근 기법을 구성하는 시그니쳐에 따른 필터링 효과를 나타낸다. 위치 시그니쳐를 통해 필터링된 후 얻은 후보 결과는 Synthetic 데이터 셋의 경우 100,000건의 시그니쳐 중에서 평균 2,181건만이 후보 결과로 남는다. 그에 반해, 위치 시그니쳐와 순서 시그니쳐에 의해 필터링되는 경우는 평균 638건만이 후보 결과로 남는다. 마지막으로 638건의 후보 시그니쳐들을 이용해서 데이터 파일을 탐색하여 실제로 데이터 파일에 질의 움직임 정보와 관련이 없는 검색 오류 매치(False Drop)를 체크한 후 최종적으로 남은 결과는 평균 627건이다.

부가 저장 공간 비율(Storage Overhead : SO)은 식 (2)와 같이 원래의 데이터 파일의 크기를 기준으로 색인 파일에서 부가적으로 더 요구되는 저장공간의 비율을 의미한다. 여기서 색인 파일의 크기는 객체 시그니쳐 파일, 움직임 시그니쳐 파일, 순서 시그니쳐 파일 그리고 링크 파일과 모두 합한 크기를 의미한다.

$$SO = \frac{\text{색인파일의크기}}{\text{데이타파일의크기}} * 100 \qquad (2)$$

SeqScan 방식의 부가 저장 공간 비율이 0%인데 비해, SAMOT 접근 기법은 Synthetic 데이터의 경우 약 16%의 부가 저장 공간을 요구하고, 대규모 리얼 데이터의 경우는 약 27% 정도의 부가 저장 공간을 요구한다.

7. 결론 및 향후 연구

본 논문에서는 비디오 데이터가 지니는 이동 객체의 궤적을 효과적으로 모델링 할 수 있는 새로운 시공간 표현 기법과 검색의 효율성을 최대화할 수 있는 이동 객체를 위한 시그니쳐 기반 접근 기법을 제안한다. 제안하는 시공간 표현 기법은 궤적을 기반으로 하는 내용 기반 검색과 궤적이 일어나는 위치 정보를 통해 얻어진 개념(의미)을 이용한 개념 기반 검색을 지원한다. 또한 제안하는 SAMOT 접근 기법은 데이터 파일을 직접 접근하기 전에 전체 시그니쳐들을 탐색하여 필터링을 수행하기 때문에 순차 탐색에 비해 많은 수의 디스크 접근 횟수를 감소시킴으로써 검색 성능을 향상시킨다. 아울러, 제안한 방법의 효율성을 측정하기 위해, 검색 효과(Retrieval Effectiveness)와 검색 효율(Retrieval Efficiency) 측면을 고려해서 성능 평가를 수행하였다. 먼저, 검색 효과 측면에서는 제안하는 방법이 타 연구인 Li와 Shan의 방법에 비해 동등한 재현율을 유지하며, 정확율 측면에서 약 20% 정도의 성능 향상을 보임을 알 수 있었다. 또한 검색 효율 측면에서는 삽입 시간의 경우, 제안하는 SAMOT 접근 기법이 SeqScan에 비해 Synthetic 데이터의 경우 약 2배 정도, 대규모 리얼 데이터의 경우 약 3배 정도가 더 소요된다. 이는 움직임 정보들을 데이터 파일에 저장하기 전에 먼저, 시그니쳐를 구성한 후, 구성된 시그니쳐 정보를 시그니쳐 파일에 저장하는 시간을 요구하기 때문이다. 검색 시간의 경우, 제안하는 SAMOT 접근 기법이 SeqScan에 비해 Synthetic 데이터의 경우 약 10배 정도, 대규모 리얼 데이터의 경우도 약 10배 정도의 성능 향상을 보인다. 이는 사용자 질의를 처리하기 위해, 데이터 파일을 직접 접근하기 전에 시그니쳐 파일을 통한 필터링 과정에서 적은 수의 후보 결과만을 토대로 데이터 파일을 접근하기 때문이다. 마지막으로, Synthetic 데이터의 경우 약 16% 정도의 부가 저장 공간을 요구하며, 대규모 리얼 데이터의 경우는 약 27% 정도의 부가 저장 공간을 요구한다.

앞으로의 연구방향은 본 시스템을 통해 사용자가 다양한 질의를 할 수 있고 또한 보다 편리하게 검색 결과를 브라우징할 수 있는 내용 및 개념 기반 축구 비디오 검색 GUI(Graphic User Interface)를 설계 및 구현하는 것이다.

참 고 문 헌

[1] W. Niblack, et. al., "The QBIC project : Quering by Iamge Content Using Color, Texture, and Shape," in Procceedings of SPIE Storage and Retrieval for Image and Video Databases, pp.173-187, 1993.

[2] J. R. Smith, S. F. Chang, "VisualSEEk : a Fully Automated Content-Based Image Query System," in Procceedings of ACM Multimedia 96, pp.87-98, 1996.

[3] T. D. C. Little, G. Ahanger, R. J. Folz, et. al., "A Digital On-Demand Video Service Supporting Content-Based Queries," in Proceedings of ACM Multimedia 93, pp.427-436, 1993.

[4] Virginia, E. Ogle and Michael Stonebraker, "Chabot : Retrieval from a Relational Database of images," IEEE Computer, Vol.28, No.9, pp.40-48, 1995.

[5] G. Ahanger, D. Benson, and T. D. C Little, "Video query formulation," in Proceedings of SPIE Electronic Imaging Science and Technology, pp.280-291, 1995.

[6] 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 International Conference on Multimedia Computing and Systems, pp.453-457, 1996.

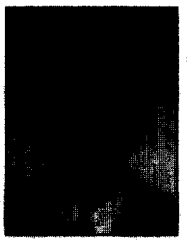
[7] John Z. Li, M. Tamer Ozsu, Duane Szafron, "Modeling Video Temporal Relationships in an Object Database Management System," in Proceedings of Multimedia Computing and Networking(MMCN97), pp.80-91, 1997.

[8] John Z. Li, M. Tamer Ozsu, Duane Szafron, "Modeling of Video Spatial Relationships in an Objectbase Management System," in Proceedings of International Workshop on Multimedia DBMS, pp.124-133, 1996.

[9] Man-Kwan Shan and Suh-Yin Lee, "Content-based Video Retrieval via Motion Trajectories," in Proceedings of SPIE Electronic Imaging and Multimedia System II, Vol.3561, pp.52-61, 1998.

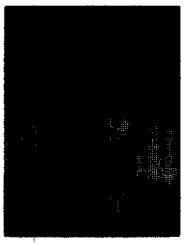
[10] Z. Aghbari, K. Kaneko, A. Makinouchi, "Modeling and Querying Videos by Content Trajectories," In Proceedings of the International Conference and Multimedia Expo, pp.463-466, 2000.

[11] C. Faloutsos and S. Christodoulakis, "Signature files : An access methods for documents and its analytical performance evaluation," ACM Transaction on Database Systems, Vol.2, No.4, pp.267-288, 1984.
[12] S. K. Chang, Q. Y. Shi and C. W. Yan, "Iconic Indexing by 2D Strings," IEEE Trans. Pattern Analysis, Machine Intelligence, Vol.9, No.3, pp.413-428, 1987.
[13] J. W. Chang, Y. J. Kim and K. J. Chang, "A Spatial Match Reprensentation Scheme Indexing and Querying in Icnoic Image Databases," ACM International Conference on Information and Knowledge Management, pp.169-176, 1997.
[14] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," Communication of the ACM, Vol.26, No.11, pp. 832-843, 1983.
[15] C. C. Chang and J. H. Jiang, "A fast spatial match retrieval using a superimposed coding technique," In Proc. of the Int's Symposium on Advanced database Technologies and Their Integration, pp.71-78, 1994.
[16] G. Salton, "A New Comparison between Conventional Indexing(MEDLARS) and Automatic Text Processing(SMART)," Journal of the American Society for Information Science, Vol.23, No.2, pp.75-84, 1972.
[17] Salton, G., M. McGill, An introduction to Modern Information Retrieval, McGraw-Hill, 1993.



심 춘 보

e-mail : cbsim@dblab.chonbuk.ac.kr
1996년 전북대학교 컴퓨터공학과(공학사)
1998년 전북대학교 컴퓨터공학과(공학석사)
2000년 전북대학교 컴퓨터공학과 박사과정 수료
관심분야 : 멀티미디어 데이타베이스, 멀티미디어 정보검색, 시공간 데이터베이스, 데이터 접근(색인) 기법 등



장 재 우

e-mail : jwchang@dblab.chonbuk.ac.kr
1984년 서울대학교 전자계산기공학과(공학사)
1986년 한국과학기술원 전산학과(공학석사)
1991년 한국과학기술원 전산학과(공학박사)
1996년~1997년 Univ. of Minnesota, Visiting Scholar.
1991년~현재 전북대학교 컴퓨터공학과 부교수
관심분야 : 멀티미디어 데이타베이스, 멀티미디어 정보검색, 고차원 색인 기법, 하부저장구조 등