이동 객체의 유사 부분궤적 검색을 위한 시그니쳐-기반 색인 기법

심 춘 보† · 장 재 우††

요 약

최근 비디오 데이타베이스, 시공간 데이타베이스, 모바일 데이타베이스와 같은 데이타베이스 응용 분야에서 이동 객체를 기반으로 하는 검색 기법에 관한 연구가 활발히 이루어지고 있다. 본 논문에서는 이동 객체의 궤적에 대한 효율적인 유사 부분궤적 검색을 지원하는 새로운 시그니쳐-기반 색인 기법을 제안한다. 제안하는 시그니쳐-기반 색인 기법은 궤적 데이타를 토대로 궤적 시그니쳐를 생성하는 방법에 따라 중첩 시그니쳐-기반 색인 기법(Superimposed signature-based Indexing scheme for similar Sub-trajectory Retrieval : SISR)과 합성 시그니쳐-기반 색인 기법(Concatenated signature-based Indexing scheme for similar Sub-trajectory Retrieval : CISR)으로 나뉜다. 생성된 궤적 시그니쳐 정보는 시그니쳐 파일에 저장되고, 검색시 주어진 사용자 질의 궤적 정보를 기반으로 데이타 파일을 직접 접근하기 전에 전체 궤적 시그니쳐들을 탐색하여 필터링을 수행한다. 이를 통해 데이타 파일의 검색 범위를 현저히 줄임으로써 검색 성능을 향상시킨다. 또한 검색된 궤적 데이타와의 유사성을 측정하기 위해 k-워핑 알고리즘을 적용시켜 검색의 효율성을 높인다. 마지막으로, 순차 색인 기법, SISR 기법, 그리고 CISR 기법을 삽입시간, 검색 시간 그리고 부가 저장 공간 측면에서 성능 평가를 수행한다. 성능 평가 결과, 제안하는 두 가지 기법이 검색 성능 측면에서 순차 색인 기법에 비해 성능이 우수함을 나타내고, 아울러 SISR 기법이 CISR 기법에 비해 보다 우수한 성능을 보인다.

Signature-based Indexing Scheme for Similar Sub-Trajectory Retrieval of Moving Objects

Choon-Bo Shim† · Jae-Woo Chang††

ABSTRACT

Recently, there have been researches on storage and retrieval technique of moving objects, which are highly concerned by user in database application area such as video databases, spatio-temporal databases, and mobile databases. In this paper, we propose a new signature-based indexing scheme which supports similar sub-trajectory retrieval as well as good retrieval performance on moving objects' trajectories. Our signature-based indexing scheme is classified into concatenated signature-based indexing scheme for similar sub-trajectory retrieval, entitled CISR scheme and superimposed signature-based indexing scheme for similar sub-trajectory retrieval, entitled SISR scheme according to generation method of trajectory signature based on trajectory data of moving object. Our indexing scheme can improve retrieval performance by reducing a large number of disk access on data file because it first scans all signatures and does filtering before accessing the data file. In addition, we can encourage retrieval efficiency by appling k-warping algorithm to measure the similarity between query trajectory and data trajectory. Finally, we evaluate the performance on sequential scan method(SeqScan), CISR scheme, and SISR scheme in terms of data insertion time, retrieval time, and storage overhead. We show from our experimental results that both CISR scheme and SISR scheme are better than sequential scan in terms of retrieval performance and SISR scheme is especially superior to the CISR scheme.

키워드 : 이동 객체(Moving Objects), 유사 부분궤적 검색(Similar Sub-Trajectory Retrieval), 시그니쳐-기반 색인 기법(Signature-based Indexing Scheme), 유사성 측정(Similarity Measure)

1. 서 론

PCS, PDA와 같은 이동 기기 보급의 확산, GPS 활용도의 급증, 그리고 무선 데이타 통신 및 무선 인터넷 기술이 발달함에 따라 이동 객체에 대한 사용자의 관심이 날로 증가하고 있다. 이에 따라 이동 객체의 위치 정보나 이동 패턴 등과 같은 움직임 정보의 효율적인 관리 및 저장을 위한 연구가 지리 정보 시스템(GIS), 시공간 데이타베이스, 그리고 비디오 데이타베이스와 같은 다양한 분야에서 활발히 이루어지고 있다. 한편, 멀티미디어 데이타 가운데 텍스트나 이미지 데이타와는 달리 비디오 데이타가 지니는 가장 중요한 특징은 이동 객체에 대한 움직임 정보이다. 이러한 움직임 정보는 각각의 프레임 내에서의 객체들 간의 공

† 준 회 원 : 부산가톨릭대학교 컴퓨터정보공학부 교수
†† 종신회원 : 전북대학교 컴퓨터공학과 교수
논문접수 : 2002년 11월 29일, 심사완료 : 2003년 6월 12일

간적인 정보와 일련의 프레임들 간의 시간적인 정보가 결합된 시공간 관계성을 통해 표현된다. 따라서 이러한 시공간 관계성은 비디오 데이타에 대한 사용자의 내용 및 개념 기반 검색을 수행하는 데 있어 매우 중요한 역할을 한다. 예를 들어, 비디오 데이타베이스에서 이동 객체의 움직임 궤적을 이용한 내용 기반 검색 질의는 다음과 같다. "사용자 인터페이스를 통해 스케치된 이동 객체의 궤적과 유사한 궤적을 가진 모든 비디오 샷(Shot)을 찾아라."

시간의 흐름에 따라 공간적 위치가 연속적으로 변하는 객체를 이동객체(moving objects)라 하며, 이러한 이동 객체의 연속적인 움직임(motion)들의 모임을 궤적(trajectories)이라 한다. 이러한 이동 객체의 궤적은 시공간 데이타베이스나 비디오 데이타베이스에서 사용자의 주된 관심의 대상이며, 내용 기반 검색을 수행하는 데 있어 매우 중요한 역할을 수행한다. 그리고 주어진 사용자 질의 궤적과 유사한 패턴을 포함하는 이동 객체의 궤적을 찾는 것을 유사 부분궤적 검색(similar sub-trajectory retrieval)이라 한다. 효율적인 유사 부분 궤적 검색을 위해서, 사용자 질의 궤적에 대해 주어진 임계값 범위 내에서 유사한 데이타 궤적을 검색할 수 있는 근사 매칭이 지원되어야 한다.

한편, 내용 기반 비디오 검색에 관한 대부분의 연구[1-5]는 비디오 데이타로부터 추출된 의미 및 내용 정보를 모델링 하는 데이타 표현 기법과 다른 미디어에 비해 큰 용량을 차지하는 비디오 데이타를 효과적으로 저장하는 데이타 저장 기법에 큰 관심을 가져왔다. 그러나 컴퓨터 하드웨어의 급속한 발달로 테라(Tera)급의 대용량 멀티미디어 데이타베이스가 일반화되고 있으며, 따라서 대용량 멀티미디어 데이타를 다루는 데 있어 주된 관심은 사용자의 다양한 질의에 대해 빠른 검색 성능을 제공할 수 있는 검색의 효율성에 있다. 아울러, 시스템을 통해 검색된 많은 후보 결과에 대해서 효율적인 유사성 함수를 통해 유사성을 측정하여 검색 결과의 질을 향상시킬 수 있는 연구가 필요하다.

따라서 본 논문에서는 이동 객체가 이루는 궤적에 대한 효율적인 저장 및 유사 부분궤적 검색을 지원하는 새로운 시그니쳐-기반 색인 기법을 제안한다. 제안하는 시그니쳐-기반 색인 기법은 궤적 데이타를 토대로 궤적 시그니쳐를 생성하는 방법에 따라 합성 시그니쳐-기반 색인 기법(Concatenated signature-based Indexing scheme for similar Sub-trajectory Retrieval : CISR)과 중첩 시그니쳐-기반 색인 기법(Superimposed signature-based Indexing scheme for similar Sub-trajectory Retrieval : SISR)으로 나뉜다. CISR 기법은 시그니쳐 파일내의 궤적 시그니쳐 레코드가 가변 길이 레코드로 구성되며, 궤적 데이타를 구성하는 움직임 요소들의 수가 가변이라는 특징과 움직임 요소들의 순서 정보가 중요하다는 특징을 고려하여 이웃한 움직임 요소들 간의 각도의 차를 이용해서 움직임 시그니쳐를 생성하고 이들을 모두 합성(concatenated)시켜 궤적 시그니쳐를 구성한다. 그에 반해, SISR 기법은 시그니쳐 파일 내의 궤적 시그니쳐 레코드가 고정길이 레코드로 구성되며, 궤적을 구성하는 첫 번째 움직임 요소에서부터 시작하여 마지막 움직임 요소까지 순회하면서 이웃한 움직임 요소들 간의 각도를 이용하여 고정된 크기의 움직임 시그니쳐를 생성하고 이들을 모두 중첩(superimposed)시켜 궤적 시그니쳐를 구성한다. 두 가지 기법 모두 생성된 궤적 시그니쳐 정보는 시그니쳐 파일에 저장되고, 검색시 주어진 사용자 질의 궤적 정보를 기반으로 데이타 파일을 직접 접근하기 전에 전체 궤적 시그니쳐들을 탐색하여 필터링을 수행한다[6, 7]. 이를 통해, 데이타 파일의 검색 범위를 현저히 줄여 검색 성능을 향상시키며, 아울러, 검색된 궤적 데이타와 질의 궤적 데이타간의 유사성을 측정하기 위해 k-워핑 알고리즘을 적용시켜 검색의 효율성을 높인다.

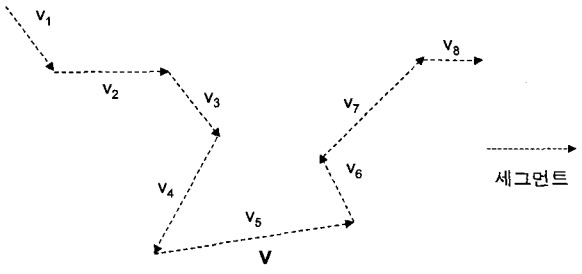
본 논문의 구성은 다음과 같다. 2장에서는 유사 부분궤적 검색과 관련된 기존 연구를 분석한다. 3장에서는 이동 객체의 효율적인 유사 부분궤적 검색을 위한 시그니쳐-기반 색인 기법을 제안한다. 4장에서는 제안하는 색인 기법의 실험 및 성능평가를 수행한다. 마지막으로, 5장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

이동 객체의 궤적과 관련된 Shan의 기법을 비롯해서 다수의 기법들과 유사 부분궤적 검색을 위한 유사성 측정 기법과 관련이 있는 유사 서브시퀀스 검색에 관한 연구를 소개한다.

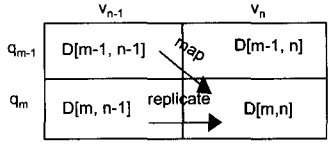
Shan[8]은 내용 기반 비디오 검색을 위해 이동 객체의 궤적을 이루는 각각의 움직임들을 (그림 1)과 같이 단일 속성인 실제 각도(0~360)로 표현하고, 사용자 질의 궤적 $Q=(q_1, q_2, \cdots, q_M)$와 유사한 부분궤적을 포함하는 데이타 궤적 $V=(v_1, v_2, \cdots, v_N)$사이의 유사성을 측정하기 위해 OCM(Optimal Consecutive Mapping)과 OCMR(Optimal Consecutive Mapping with Replication)의 두 가지 유사성 측정 알고리즘을 제안하였다.

V = (310°, 0°, 310°, 240°, 5°, 95°, 45°, 0°)



(그림 1) 예제 움직임 경로

먼저, OCM 알고리즘은 정확 매칭으로 질의 궤적과 데이타 궤적사이의 각 움직임들 간에 일대일 매핑으로 유사성을 측정한다. 즉, 질의 궤적 Q를 이루는 움직임들을 데이타 궤적 V의 첫 번째 움직임(v_1)으로부터 질의 궤적의 움직임 수(M)만큼 순서대로 매핑하여 유사성을 측정한다. 마찬가지로, 데이타 궤적의 두 번째 움직임(v_2)으로부터 M만큼 순서대로 매핑해서 유사성을 측정한다. 이렇게 구한 유사성들 중에서 가장 작은 값을 질의 궤적 Q와 데이타 궤적 V사이의 유사성으로 선택한다. OCMR 알고리즘은 근사 매칭을 지원하며, (그림 2)에서와 같이 사용자로부터 주어진 질의 궤적을 이루는 각각의 움직임들(q_i)과 데이타 궤적을 이루는 각각의 움직임들(v_j) 사이에 유사성을 계산하는데 있어, 질의 움직임들 중에서 자기 자신을 반복시킨 것(replicate)과 그렇지 않은 것(map)을 비교해서 더 작은 것을 선택한다. 여기서 $d(q_i, v_j)$는 q_i와 v_j사이의 거리 함수(distance function)이고, D[M, N]는 질의 궤적과 데이타 궤적사이의 반복(replication)을 이용한 최소 거리(minimum distance)를 구하기 위한 테이블이다.



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

(그림 2) D[m, n]과 D[i, j]사이의 OCMR의 관계

Shan의 연구에서는 이동 객체의 모델링을 하는데 있어 다지 방향 정보만을 고려하고 있으며 거리 정보는 고려하고 있지 않다. 또한 유사 부분궤적 검색을 위해 질의 궤적과 데이터 궤적 사이의 유사성을 측정하는데 있어 데이터 궤적을 구성하는 움직임 요소에 대해서 무한히 반복을 허용해서 유사성을 측정한다. 이는 비디오 데이터내의 이동 객체의 궤적의 특성상 검색의 재현율을 떨어뜨린다. 마지막으로, 대용량의 이동 객체의 궤적 데이터를 위한 효율적인 색인 기법에 관한 연구가 아직까지 이루어지고 있지 않다.

비디오 데이터 내의 이동 객체의 궤적에 대한 Shan의 기법 이외의 기타의 연구들 중에서 Li[9]는 이동 객체의 궤적을 모델링하기 위해 8가지의 방향으로 코드화해서 표현하고 있으며, 두 궤적 사이의 유사성 측정을 위한 유사성 측정 함수를 제공하고 있다. Shepherd[10]는 이동 객체의 궤적을 위해 기존의 객체간의 공간 관계성을 이용해 이미지 검색 기법에 사용했던 방법인 2D-PIR 표현 방법을 확장하여 ST-PIR(Spatial-Temporal Projection Interval Relationships) 기법을 제안하였다. 마지막으로, Dagtas[11]는 비디오 데이터 내의 이동 객체의 궤적을 이용하여 검색 및 색인을 수행할 수 있는 PICTURESQUE라는 시스템을 구현하였다.

유사 서브시퀀스(subsequence) 검색[12-14]은 주어진 질의 시퀀스를 포함하는 유사한 데이타 시퀀스를 검색하는 것으로 주가 데이타, 상품 판매량, 날씨 데이타, 의료 데이타와 같은 응용 분야에서 많은 연구가 이루어졌다. 우선 시퀀스 데이타베이스는 다양한 길이의 시퀀스들로 구성되며, 시퀀스 S(=<s[1], s[2], …, s[|S|]>)는 일정한 시간 주기마다 얻어진 연속된 실수 값들로 이루어진다. 여기서 |S|는 시퀀스의 길이이다. s[i]는 S의 i번째 요소를 나타내며, s[i : j]는 i번째 위치에서 j번째 위치를 포함하는 서브시퀀스를 의미한다. s[i : -]는 i번째 위치에서 시작해서 시퀀스 S의 마지막 요소까지를 나타낸다. ( )는 요소가 없는 널 시퀀스(null sequence)를 의미한다.

주어진 질의 시퀀스와 데이타 시퀀스를 구성하는 각 요소들 간의 유사성을 측정하기 위해서 식 (1)과 같은 거리 함수를 사용한다.

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

L_1은 맨하탄 거리(manhatann distance), L_2는 유클리드 거리(euclidean distance), L_∞는 대응되는 각 쌍의 거리 중 최대 거리를 의미한다.

효율적인 유사 서브시퀀스 검색을 위해 정규화(normalization), 이동 평균(moving average), 타임 워핑(time warping)등의 다양한 알고리즘들이 제안되었다. 그 중에서도 특히 타임 워핑 기법은 시퀀스내의 각 요소 값을 임의의 수만큼 반복시키는 것을 허용하는 알고리즘이다. 이를 통해 사용자의 부정확한 대략적인 질의에 대해서 어느 정도의 허용치 범위 내에서 사용자의 질의를 변형함으로써 사용자가 원하는 검색 결과를 보장하는 근사 매칭을 지원한다. 다음은 유사 서브시퀀스 검색을 위해 제안된 타임 워핑 거리를 나타낸다.

$$D_{tw}(( ), ( )) = 0$$

$$D_{tw}(S, ( )) = D_{tw}(( ), Q) = \infty$$

$$D_{tw}(S, Q) = D_{base}(S[1], Q[1]) + \min(D_{tw}(S, Q[2 : -]), D_{tw}(S[2 : -], Q), D_{tw}(S[2 : -], Q[2 : -])) \qquad (2)$$

$$D_{base}(a, b) = |a - b|$$

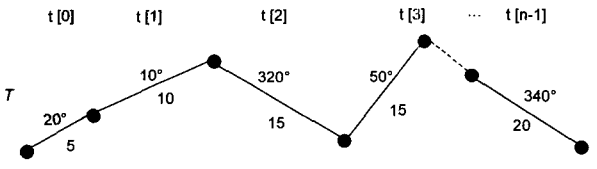
타임 워핑 변환을 이용한 유사 서브시퀀스 검색은 식 (2)에서와 같이 질의 시퀀스와 데이타 시퀀스를 구성하는 임의의 요소에 대해서 임의의 수만큼 제한없이 반복시키는 것을 허용한다.

3. 유사 부분궤적 검색을 위한 시그니쳐-기반 색인 기법

본 논문에서는 이동 객체가 이루는 궤적에 대한 효율적

인 유사 부분궤적 검색을 지원하며, 아울러 주어진 사용자 질의 궤적에 대해서 빠른 검색 성능을 제공해 줄 수 있는 시그니쳐-기반 색인 기법을 제안한다. 제안하는 시그니쳐-기반 색인 기법은 이동 객체의 궤적 데이타를 이용하여 시그니쳐 파일의 궤적 시그니쳐 레코드를 생성하는 방식에 따라 합성 시그니쳐-기반 색인 기법(Concatenated signature-based Indexing scheme for similar Sub-trajectory Retreival : CISR)과 중첩 시그니쳐-기반 색인 기법(Superimposed signature-based Indexing scheme for similar Sub-trajectory Retrieval : SISR)으로 나뉜다. CISR 기법은 시그니쳐 파일내의 궤적 시그니쳐 레코드가 가변 길이 레코드로 구성되며, 궤적 데이타를 구성하는 움직임 요소들의 수가 가변이라는 특징과 움직임 요소들의 순서 정보가 중요하다는 특징을 고려하여 이웃한 움직임 요소들 간의 각도의 차를 이용해서 움직임 시그니쳐를 생성하고 이들을 모두 합성시켜 궤적 시그니쳐를 구성한다. 그에 반해, SISR 기법은 시그니쳐 파일 내의 궤적 시그니쳐 레코드가 고정길이 레코드로 구성되며, 궤적을 구성하는 첫 번째 움직임 요소에서부터 시작하여 마지막 움직임 요소까지 순회하면서 이웃한 움직임 요소들 간의 각도를 이용하여 고정된 크기의 움직임 시그니쳐를 생성하고 이들을 모두 중첩시켜 궤적 시그니쳐를 구성한다. 따라서 제안하는 CISR 기법과 SISR 기법 모두 생성된 궤적 시그니쳐 정보는 시그니쳐 파일에 저장되고, 검색 시 주어진 사용자 질의 궤적 정보를 기반으로 데이타 파일을 직접 접근하기 전에 전체 궤적 시그니쳐들을 탐색하여 필터링을 수행한다. 이를 통해, 데이타 파일 접근시 요구되는 디스크 접근 횟수를 현저히 줄임으로써 검색 성능을 향상시킬 수 있으며, 아울러 검색된 궤적 데이타와 질의 궤적 데이터 간의 유사성을 측정하기 위해 방향뿐만 아니라 거리 정보까지 고려하는 k-워핑 알고리즘을 적용시켜 검색의 효율성을 높인다.

본 논문에서는 이동 객체의 궤적 T를 구성하는 움직임 요소(t[i])에 대해서 (그림 3)과 같이 방향(direction) 정보와 거리(distance) 정보를 이용하여 모델링 한다.



(그림 3) 이동 객체의 궤적 T

3.1 합성 시그니쳐-기반 색인 기법(CISR)

제안하는 CISR 색인 기법에서 이동 객체의 궤적 데이타에 대한 궤적 시그니쳐를 생성하는 방법은 궤적을 구성하는 각 움직임들 간의 움직임 시그니쳐를 생성한 후에, 생성된 모든 움직임 시그니쳐를 합성시켜 궤적 시그니쳐를 생성한다. 생성하는 방법은 다음과 같다. 먼저, 주어진 이동 객체의 궤적 정보에서 움직임 정보를 위해 방향 정보만을 추출하여 방향 궤적 T_{ang}을 구성한다.

$$T_{ang} = \{ a_i \mid i=0, \cdots, n-1\}, \ n = \mid T_{ang} \mid \quad (3)$$

여기서, ai는 방향 궤적 T_{ang}를 구성하는 i번째 움직임 요소의 방향 정보를 나타내며 실제 각도[0, 360]를 이용하여 표현한다. n(=|T_{ang}|)은 궤적을 구성하는 움직임의 수를 의미한다. 궤적을 구성하는 임의의 움직임 요소 a_i와 a_{i+1}에 대해서 ADF(Angle Difference Function) 함수를 이용하여 두 움직임 요소 사이의 각도의 차(Angle Difference : AD)를 계산한다.

$$AD_i = ADF(a_i, \ a_{i+1}), \ i=0, \cdots, n-2 \quad (4)$$

```
int ADF(int a1, int a2)
{
    int Ang_Diff ;
    Ang_Diff = 180° + (a2 - a1) ;
    if( Ang_Diff > 360° ) Ang_Diff = Ang_Diff - 360° ;
    else if(Ang_Diff < 0) Ang_Diff = Ang_Diff + 360° ;
    return Ang_Diff ;
}
```

위에서 구한 i번째 각도의 차인 AD_i를 통해 방향 코드값(direction code) dc_i를 구한다.

$$dc_i = AP(AD_i), \ i=0, \cdots, n-2 \quad (5)$$
$$= \lfloor AD_i / U_{ang} \rfloor + 1$$

여기서, AP(x)는 x에 대한 방향 코드값을 구하는 함수로 x를 단위 각도(U_{ang})로 나눈 후에 얻은 몫에 1을 더해서 구한다. 이것을 기반으로 CISR 기법을 위한 움직임 시그니쳐(motion signature : cm_sig)를 생성한다. 방향 궤적 T_{ang}를 구성하는 움직임 요소들에 대해서 i번째 움직임 시그니쳐, cm_sig_i는 식 (6)으로부터 구한다. 하나의 움직임 시그니쳐를 위한 비트 수 $m = \lceil \log_2^p \rceil + 1$, $p = \lceil 360/U_{ang} \rceil$이다.

$$cm\_sig_i = C\_Bin(dc_i, \ m), \ i=0, \cdots, n-2 \quad (6)$$

여기서, C_Bin(i, j)는 정수 i에 대해서 j비트 수만큼을 할당한 후, 이를 이진수로 변환하는 함수이다. 따라서, 이동 객체의 궤적 T에 대한 CISR 기법을 위한 궤적 시그니쳐 T_{sig}는 식 (7)과 같이 구해진 움직임 시그니쳐들을 합성시키는 연산자(⊙)를 통해 합성(concatenate)시켜 생성한다.

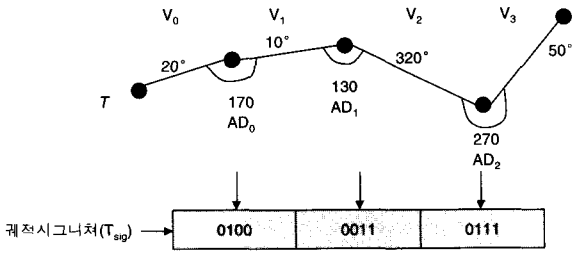
$$T_{sig} = \{cm\_sig_0 \odot cm\_sig_1 \odot \cdots \odot cm\_sig_{n-2}\} \quad (7)$$

예를 들어, (그림 4)와 같이 궤적 데이타 T의 움직임 요소 중에 각도가 {20°, 10°, 320°, 50°}일 때, 움직임 요소들 간의 각도의 차인 AD는 식 (4)를 이용하여 각각 170°,

130°, 270°를 구한다. 구해놓은 AD값들을 토대로 (그림 5)와 같이 단위 각도가 45°일 때 각각의 방향 코드값들은 '4', '3', '6'이다. 여기서 움직임 시그니쳐를 위한 비트 수 m은 다음과 같이 계산된다.

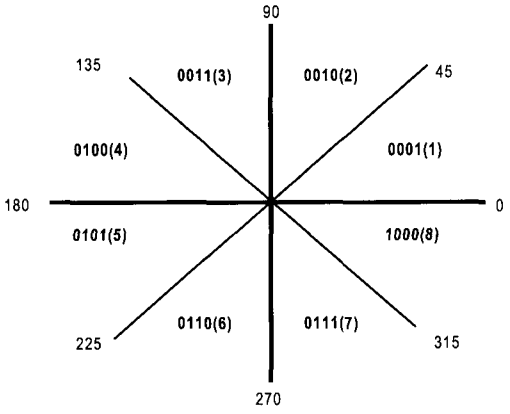
$$m = \lceil \log_2 p \rceil + 1,\ p = \lceil 360/U_{ang} \rceil$$
$$= \lceil \log_2 8 \rceil + 1,8 = \lceil 360/45 \rceil$$
$$= 4$$

따라서 방향 코드값과 움직임 시그니쳐를 위한 비트 수 m을 이용하여 각각의 움직임 시그니쳐를 구하면 '0100', '0011', '0111'이 된다. 최종적으로 궤적 T에 대한 궤적 시그니쳐 T_{sig}는 식 (7)를 통해 구해놓은 모든 움직임 시그니쳐를 합성시켜 '0100 0011 0111'로 표현한다.



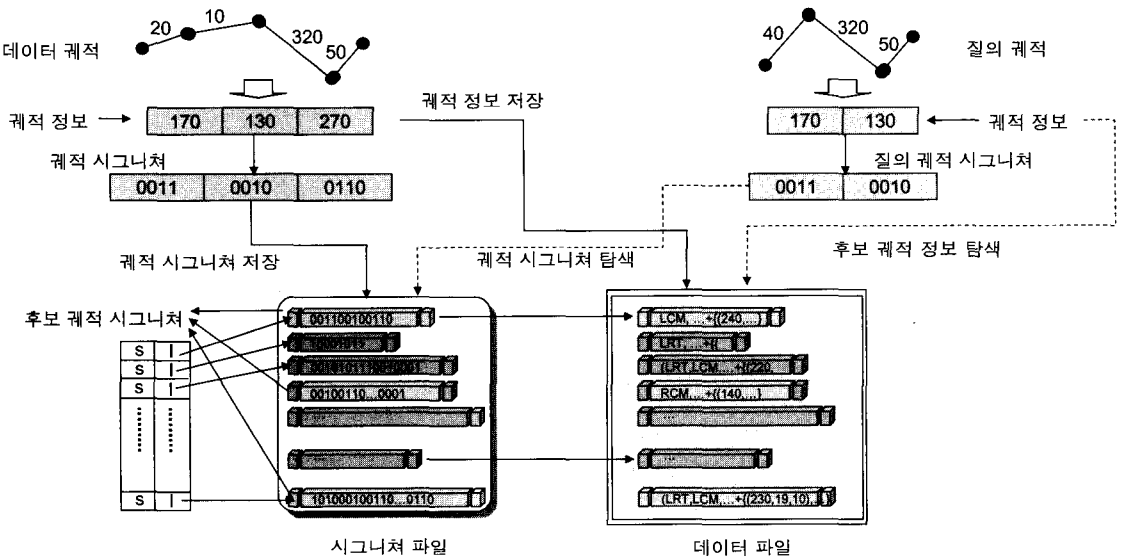
(그림 4) CISR 기법에서 AD를 이용한 궤적 시그니쳐 생성의 예

움직임 시그니쳐의 비트 수는 (그림 5)에서와 같이 각도를 나누는 단위 즉, 단위 각도에 따라 다르며, 적용하려는 응용 분야와 검색 성능에 중요한 인자이다. 즉, 단위 각도가 작을수록 움직임 시그니쳐의 비트 수는 늘어나게 되며, 질의 처리 시에 시그니쳐의 필터링 효과가 높아지기 때문에 검색의 정확성은 향상될 수 있지만, 시그니쳐 파일의 부가저장 공간의 오버헤드가 증가된다. 이렇게 생성된 궤적 시그니쳐는 궤적 데이타를 이루는 각 움직임 요소들 간의 순서 정보를 유지할 수 있을 뿐만 아니라, 질의 처리 시 상대적으로 크기가 작은 시그니쳐 정보를 접근함으로써 검색 성능을 향상시킬 수 있다.



(그림 5) 단위 각도(Uang) = 45도일 때의 움직임 시그니쳐

제안하는 CISR 색인 기법은 크게 세 개의 파일로 구성된다. 즉, 축구 비디오 샷으로부터 추출된 이동 객체(축구공)의 궤적 데이타들을 저장하고 있는 데이타 파일(*.trj), 궤적 데이타에 대한 궤적 시그니쳐를 생성하여 저장하는 시그니쳐 파일(*.sig), 그리고 시그니쳐 파일내의 궤적 시그니쳐 정보가 가변이기 때문에 각각의 시그니쳐에 대한 시그니쳐 파일 내에서의 시작 위치(Offset)와 크기 정보를 유지하는 시그니쳐 링크 파일(*.slk)이다. 본 논문에서 제안하는 CISR 색인 기법의 전체적인 구조는 (그림 6)과 같다.



(그림 6) 합성 시그니쳐-기반 색인 기법(CISR)의 전체 구조

주어진 이동 객체의 궤적 데이타를 저장하는 삽입 과정은 다음과 같다. 먼저, 궤적 데이타의 움직임 요소로부터 각도를 추출하여 이를 토대로 움직임 시그니쳐를 생성하고 생성된 모든 움직임 시그니쳐를 결합하여 궤적 시그니쳐를 만든다. 그런 다음, 주어진 궤적 데이타를 데이타 파일에 삽입한다. 데이타 파일에서의 궤적 데이타의 저장 위치(Offset)와 크기 정보를 미리 생성된 궤적 시그니쳐와 결합하여 시그니쳐 파일에 삽입한다. 마지막으로 시그니쳐 파일에서의 궤적 시그니쳐의 시작 위치와 크기 정보를 시그니쳐 링크 파일에 삽입하는 것으로 삽입 과정을 마친다.

이동 객체의 궤적을 이용한 사용자의 질의 처리는 다음과 같다. 먼저 질의 궤적이 주어지면, 질의 궤적의 각도 정보를 이용해서 삽입 과정에서 데이타 궤적 시그니쳐를 생성할 때와 같은 궤적 시그니쳐 생성 알고리즘을 이용해서 질의 궤적 시그니쳐를 생성한다. 이렇게 생성된 질의 궤적 시그니쳐를 사용하여 시그니쳐 파일에 저장된 각각의 데이타 궤적 시그니쳐들을 순차적으로 탐색하여 필터링을 수행한다. 시그니쳐 필터링 단계에서는 질의 궤적 시그니쳐와 데이타 궤적 시그니쳐간의 거리의 차가 δ이하인 값을 가지는 것만을 후보 시그니쳐로 선택한다. (그림 7)은 질의 궤적 시그니쳐와 데이타 궤적 시그니쳐를 입력 받아 후보 시그니쳐인지 아닌지를 판별하는 함수이다.

(그림 8)에서와 같이 주어진 질의 궤적 시그니쳐 Qsig를 이용하여 각각 데이타 궤적 시그니쳐 T_1sig, T_2sig, T_3sig에 대해서 (그림 7)의 후보 시그니쳐 판별함수를 적용해 보면 δ값을 2로 했을 경우, T_1sig, T_2sig는 후보 시그니쳐로 선택되며, T_3sig는 δ보다 더 크므로 후보 시그니쳐에서 제외된다. 따라서 이렇게 시그니쳐 탐색을 통해 최종적으로 선택된 후보 시그니쳐들만을 가지고 실제 궤적 데이타를 얻기 위해 데이타 파일을 탐색하게 된다. 마지막으로 데이타 파일을 탐색해서 얻은 데이타 궤적 정보와 주어진 질의 궤적 사이의 유사성을 측정한다. 이 과정에서는 보다 정확하고 효율적인 유사성 측정을 위해서 k-워핑 거리 알고리즘을 적용한다. k-워핑 거리 알고리즘에 대한 자세한 내용은 3.3절에서 기술한다.

```
int is CandidateSig(T_Sig, Q_Sig)
{
   diff_dist   // difference of distance between T_Sig and Q_Sig
   NT          // number of T_Sig
   NQ          // number of Q_Sig
    δ          // threshold value

   for(i=0 ; I < NQ-NT+1 ; i++) {
      dist_value = 0 ;
      for(j=0 ; j<NQ ; j++) {
            dist_value + = |T_Sigi+j - Q_Sigj| ;
      }
      if(dist_value < = δ)
            return CANDIDATE ;
   }

   return DISCARD ;
}
```

(그림 7) CISR 색인 기법에서 후보 시그니쳐를 판별하는 함수

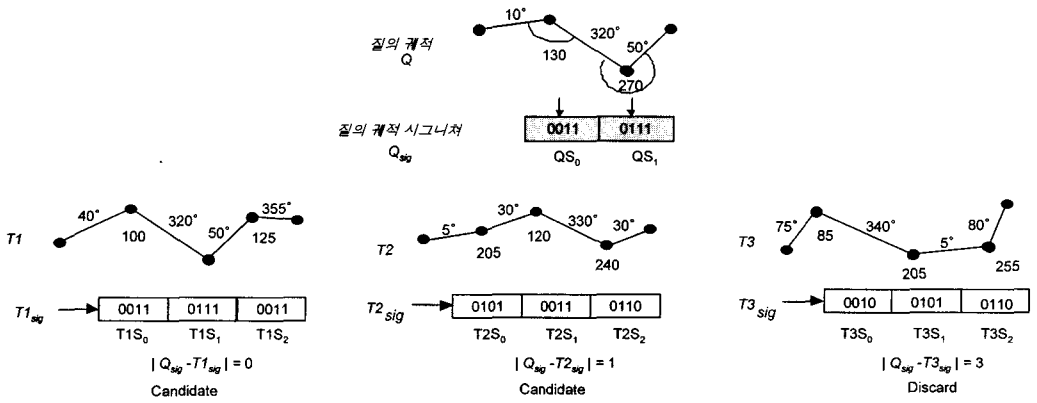
3.2 중첩 시그니쳐-기반 색인 기법(SISR)

제안하는 SISR 색인 기법에서 이동 객체의 궤적 데이타에 대한 궤적 시그니쳐를 생성하는 방법은 CISR 색인 기법에서는 달리 궤적을 구성하는 각 움직임들 간의 움직임 시그니쳐를 생성한 후에, 생성된 모든 움직임 시그니쳐를 중첩(superimpose)시켜 궤적 시그니쳐를 생성한다. 생성하는 방법은 다음과 같다. 먼저, 주어진 이동 객체의 궤적 정보에서 방향 정보만을 추출하여 각도 정보로만 이루어진 방향 궤적 T_{ang}을 구성한다.

$$T_{ang} = \{ a_i | i=0, \ldots, n-1 \}, \; n = | T_{ang} | \tag{8}$$

여기서, a_i는 방향 궤적 T_{ang}의 i번째 각도를 의미하며, $|T_{ang}|$는 궤적을 구성하는 움직임의 수를 나타낸다. 궤적을 구성하는 임의의 각도 a_i에 대해서 식 (9)를 이용하여 각도에 대한 방향 코드값(direction code : dc)을 구한다.

$$\begin{aligned} dc_i &= AP(a_i) \\ &= (a_i / U_{ang}) + 1 \end{aligned} \tag{9}$$



(그림 8) CISR 색인 기법에서 후보 시그니쳐를 선별하는 예

여기서, AP(x)는 x에 대한 방향 코드값을 구하는 함수로 x를 단위 각도(Uang)로 나눈 후에 1을 더해서 구한다. 위에서 구한 방향 코드값 dc를 이용하여 방향 궤적 Tang의 첫 번째 움직임 요소부터 마지막 움직임 요소까지 순서대로 두 개의 움직임 요소들끼리 묶어 놓은 것을 pdc(pair of dc)라고 하며, 이것을 기반으로 SISR 색인 기법에서의 움직임 시그니쳐(motion signature : sm_sig)를 생성한다.

$$pdc_i = (dc_i, dc_{i+1}),\ i = 0, \cdots, n-2 \qquad (10)$$

여기서, dc_i와 dc_{i+1}는 각각 i번째 움직임 시그니쳐(sm_sigi)에서 '1'로 세팅시킬 비트의 위치를 결정하기 위한 인덱스(index)와 오프셋(offset) 역할을 담당한다. 방향 궤적 T_{ang}를 구성하는 움직임 요소들에 대해서 i번째 움직임 시그니쳐, sm_sig_i 는 식(11)로 부터 구한다. 움직임 시그니쳐의 전체 비트수 b = $(360/U_{ang})^2$이다.

$$\begin{aligned} sm\_sig_i &= S\_Bin(d_{ci}, dc_{i+1}),\ i=0, \cdots, n-2 \\ &= 2^{k-1},\ k = (dc_i - 1) * (360/U_{ang}) + dc_i + 1 \end{aligned} \qquad (11)$$

따라서, 방향 궤적 T_{ang}에 대한 궤적 시그니쳐 T_{sig}는 식(12)와 같이 구해진 움직임 시그니쳐들을 모두 중첩(superimpose)시켜 즉, bit-wise Oring 연산(⊗)을 이용해 생성한다.

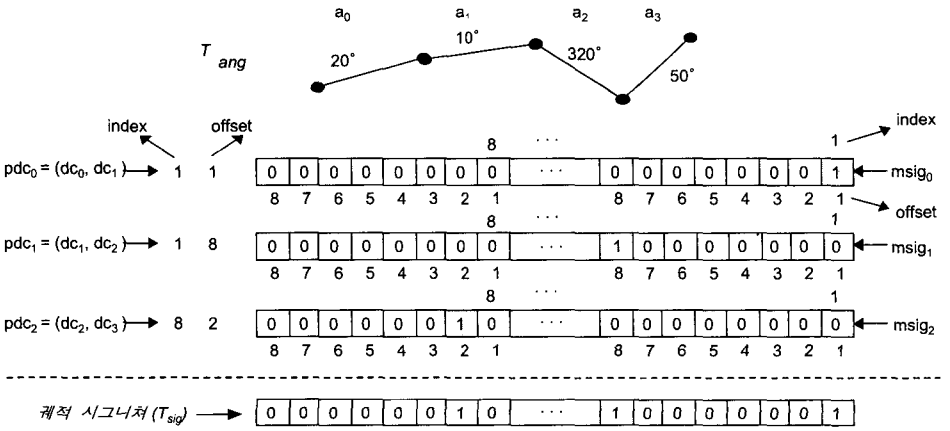
$$T_{sig} = \{ smsig_0 \otimes smsig_1 \otimes \cdots \otimes smsig_{n-2} \} \qquad (12)$$

예를 들면, 각도 궤적 T_{ang}가 {20°, 10°, 320°, 50°}일 때, 단위 각도(U_{ang})를 45°라고 할 때, 각각의 각도에 대한 방향 코드값 dc는 식 (9)에 의해 {1, 1, 8, 2}가 된다. 즉, 방향 코드값 dc는 (그림 5)에서와 같이 360°를 단위 각도 45°로 나누어 각각 방향 코드를 할당한 것으로, 단위 각도(U_{ang})는 응용 분야에 적합한 값으로 설정한다. 첫 번째 움직임 요소(a_0)에서부터 시작해서 마지막 움직임 요소(a_3)까지 순서대로 두 개의 움직임 요소들끼리 묶은 pdc_0, pdc_1, pdc_2 즉, (1, 1), (1, 8), (8, 2)을 이용하여 각각의 움직임 시그니쳐 msig를 생성한다. 각각의 움직임 시그니쳐 msig는 움직임 요소의 각도에 대한 방향 코드값(dc)이 1부터 8까지이므로 pdc는 실제로 (1, 1)부터 (8, 8)까지 총 64가지 중에 하나가 될 수 있다. 따라서 움직임 시그니쳐는 64비트를 할당해서 그 중에 한 비트를 '1'로 세팅함으로써 생성하고 '1'로 세팅될 비트의 위치는 pdc의 첫 번째 요소와 두 번째 요소를 각각 인덱스(index)와 오프셋(offset)으로 결합하여 정한다. 마지막으로 궤적 시그니쳐 T_{sig}는 구해 놓은 3개의 움직임 시그니쳐들을 중첩 코딩 방식(bit-wise Oring)을 이용하여 구한다. (그림 9)는 이동 객체의 궤적 정보에 대한 SISR 색인 기법에서의 궤적 시그니쳐를 생성한 예를 보인다.

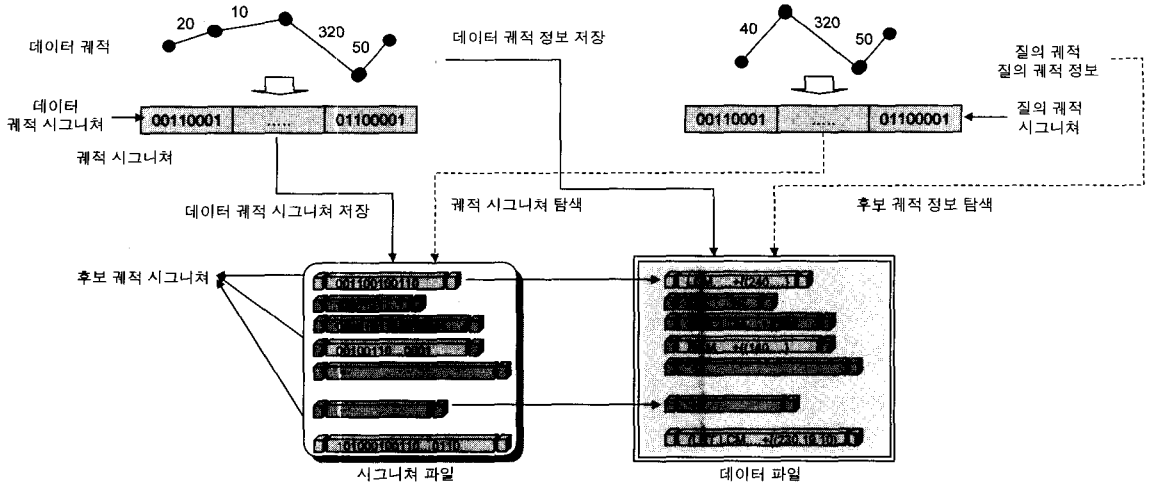
SISR 색인 기법에서 궤적 시그니쳐는 중첩 방식을 이용해서 생성하기 때문에 CISR 색인 기법의 합성 방식을 이용한 궤적 시그니쳐 생성 방식에서 얻을 수 있는 움직임 요소들 간의 순서 정보를 유지할 수 없는 단점이 있다. 그러나 궤적 데이타를 구성하는 움직임 요소의 수가 가변이더라 하더라도 항상 고정된 크기를 갖는 궤적 시그니쳐 레코드를 생성할 수 있으며, 또한 질의 궤적 시그니쳐와 데이타 궤적 시그니쳐간의 매칭 시 비트 연산을 수행할 수 있기 때문에 시그니쳐 파일 탐색 시간이 매우 줄일 수 있는 장점을 가진다.

SISR 색인 기법은 CISR 색인 기법과는 달리, 축구 비디오 샷으로부터 추출된 이동 객체(축구공)의 궤적 데이타들을 저장하고 있는 데이타 파일(*.trj)과 궤적 데이타에 대한 고정된 크기의 궤적 시그니쳐를 생성하여 저장하는 시그니쳐 파일(*.sig) 두 파일로 구성된다. (그림 10)은 본 논문에서 제안하는 SISR 색인 기법의 전체적인 구조를 나타낸다.

주어진 이동 객체의 궤적 데이타를 저장하는 삽입 과정은 CISR 색인 기법과 거의 유사하다. 단, 이동 객체의 궤적을 이용한 사용자의 질의 처리는 CISR 방법과 약간 다르다. 먼저 질의 궤적이 주어지면, 질의 궤적의 각도 정보를 이용해서 삽입 과정에서 데이타 궤적 시그니쳐를 생성할 때와 같은 궤적 시그니쳐 생성 알고리즘을 이용해서 질



(그림 9) SISR 색인 기법에서 궤적 시그니쳐를 생성하는 예



(그림 10) 중첩 시그니쳐-기반 색인 기법(SISR)의 전체 구조

의 궤적 시그니쳐를 생성한다. 이렇게 생성된 질의 궤적 시그니쳐를 사용하여 시그니쳐 파일에 저장된 각각의 데이타 궤적 시그니쳐들을 순차적으로 탐색하여 필터링을 수행한다. 시그니쳐 필터링 단계에서는 질의 궤적 시그니쳐를 포함하는 데이타 궤적 시그니쳐만을 시그니쳐로 선택한다. 마지막으로 데이타 파일을 탐색해서 얻은 데이타 궤적 정보와 주어진 질의 궤적 사이의 유사성을 측정한다. 이 과정에서는 CISR 색인 기법과 같은 방법을 이용한다.

본 논문에서는 이동 객체의 궤적을 모델링하기 위해 방향과 거리를 모두 고려한다. 그러나 제안하는 CISR 기법이나 SISR 기법 모두 방향 정보만을 이용하여 궤적 시그니쳐를 생성한 후 이를 이용하여 1차 검색 즉, 질의 궤적과 관련이 없는 불필요한 데이터를 필터링하는 과정을 거친다. 이 때 필터링에서 얻는 데이터 궤적들에 대해서만 방향과 거리를 모두 고려하여 2차 검색을 수행하면서 질의 궤적과의 유사성을 측정한다.

3.3 유사 부분궤적 검색을 위한 유사성 측정

시그니쳐 필터링 단계에서 구한 후보 시그니쳐에 상응하는 데이타 궤적과 질의 궤적 사이의 유사성 측정은 효율적인 유사 부분궤적 검색을 위한 k-워핑 거리 알고리즘(k-warping distance algorithm)을 이용한다. k-워핑 거리 알고리즘은 시계열 데이타베이스에서 유사 서브시퀀스 검색을 위해 제안되었던 타임 워핑(time-warping) 거리 알고리즘을 이동 객체의 궤적 데이타에 맞게 변형한 알고리즘이다. 즉, 타임 워핑 거리 알고리즘은 시계열 데이타의 특성상 주어진 질의 시퀀스뿐만 아니라 데이타 시퀀스에 대해서도 무한 반복을 허용함으로써 유사 서브시퀀스 검색의 효율성을 향상시킨 알고리즘이다. 그러나 이동 객체의 궤적 데이타와 시계열 데이타와는 데이타의 특성이 매우 상이하기 때문에 효율적인 유사 부분궤적 검색을 위해서는 무한 반복을 허용하는 타임 워핑 거리 알고리즘을 그대로 적용시킬 수 없다. 따라서 본 논문에서는 이동 객체의 유사 부분궤적 검색의 효율성을 위해, 무한 반복을 허용하는 타임 워핑 알고리즘을 변형시켜 임의의 k번까지만 반복을 허용하도록 하는 k-워핑 거리 알고리즘을 이용하여 두 궤적 사이의 유사성을 측정한다. 여기서, 움직임 요소에 대한 반복 횟수 k는 유사 부분궤적 검색을 적용하려는 응용 분야의 데이타 도메인의 특성에 따라 유동적일 수 있다. 예를 들면, 축구, 농구, 하키등과 같은 스포츠 비디오 데이타 내의 축구공(ball)이나 하키퍽(puck)과 같은 주요 이동 객체를 위한 유사 부분궤적 검색 시에는 궤적을 이루는 움직임 개수가 적기 때문에 k를 낮게 정하는 것이 검색 성능을 저하시키기 않는다. 그에 반해, PCS나 PDA와 같은 휴대용 단말기 추적과 같은 응용 분야의 경우는 궤적을 구성하는 움직임 요소의 개수가 많기 때문에 k를 높게 설정하는 것이 좋다.

질의 궤적 Q(={q[1], q[2], …, q[|Q|])과 데이타 궤적 S(={s[1], s[2], …, s[|S|])사이의 유사성을 측정하기 위해서는 먼저, 질의 궤적을 구성하는 하나의 움직임 요소 q[i]와 데이타 궤적을 구성하는 하나의 움직임 요소 s[j] 사이의 유사성을 측정할 수 있는 거리 함수, d_{df}(Q, S)를 정의하다. 여기서 q[i], s[j]는 모두 각도(angle)와 거리(distnace)의 쌍으로 이루어진다. q[i, 1]과 q[i, 2]는 질의 궤적의 i번째 움직임 요소의 각도와 거리를 의미한다. d_{dis}와 d_{ang}는 각각 움직임 요소간의 각도와 거리를 계산하는 거리 함수를 나타낸다. 식 (13)은 움직임 요소를 구성하는 각도와 거리를 모두 고려하여 결합한 거리 함수를 나타내며, α와 β는 각각 각도와 거리를 위한 가중치를 의미한다.($\alpha+\beta=1.0$)

$$
\begin{aligned}
&d_{dis}(q[i,2],s[j,2]) = |q[i,2]-s[j,2]| \\
&\text{if } |q[i,1]-s[j,1]| > 180 \ \textit{Then} \\
&\quad d_{ang}(q[i,1],s[j,1]) = 360-(|q[i,1]-s[j,1]|) \textit{else} \\
&\quad d_{ang}(q[i,1],s[j,1]) = |q[i,1]-s[j,1]| \\
&d_{df}(q[i],s[j]) = ((d_{ang}/180)*\alpha)+((d_{dis}/100)*\beta)
\end{aligned}
\qquad (13)
$$

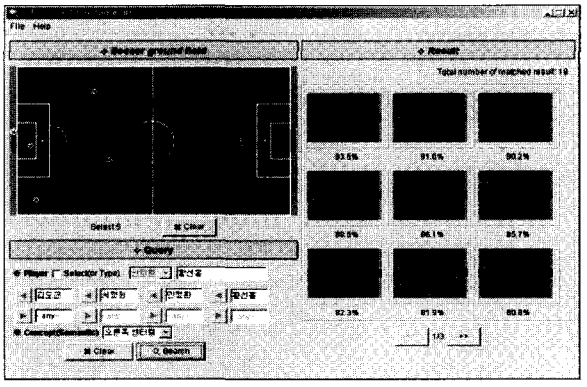
따라서 질의 궤적 Q와 데이타 궤적 S사이의 유사성을 측정하기 위한 k-워핑 거리 $D_{kw}(S, Q)$는 다음과 같이 재귀적인 호출에 의해서 구해진다.

$$
\begin{aligned}
&D_{kw}(0,\ 0) = 0\\
&D_{kw}(S,\ 0) = D_{kw}(0,\ Q) = 0\\
&D_{kw}(S,\ Q) = D_{base}(s[1],\ q[1]) +\\
&\qquad \min(\{\ D_{kw}(s[2+i:\ -],\ Q),\ 0 \le i < k\},\\
&\qquad D_{kw}(s[2:\ -],\ q[2:\ -]))\\
&D_{base}(a,\ b) = d_{df}(a,\ b)
\end{aligned}
\tag{14}
$$

여기서, S와 Q는 각각 데이타 궤적과 질의 궤적을 의미하며, s[i]와 q[j]는 각각 데이타 궤적에서의 i번째 움직임 요소와 질의 궤적에서의 j번째 움직임 요소를 의미한다. 그리고 s[2 : -]와 q[2 : -]는 각각 데이타 궤적과 질의 궤적의 2번째 움직임 요소에서 마지막 요소까지를 의미한다.

4. 실험 및 성능 평가

유사 부분궤적 검색을 위해 제안한 시그니쳐-기반 색인 기법의 효율성을 검증하기 위해 512MB 메인 메모리를 탑재한 펜티엄 Ⅳ-1.7G, 윈도우즈 2000 서버 운영체제 환경하에서 성능 평가를 수행한다. 아울러, 축구 비디오 데이타베이스를 기반으로 원하는 축구 비디오 샷을 검색하기 위한 사용자의 질의(query)를 보다 쉽고 편리하게 생성할 수 있는 사용자 인터페이스로 자바(Java) 언어를 이용하여 플랫폼에 독립적으로 구동될 수 있도록 구현한다. (그림 11)은 축구 비디오 검색 GUI로 축구 비디오 샷을 검색할 수 있도록 질의를 생성하는 질의 생성 부분과 해당 질의에 만족하는 검색 결과를 브라우징 할 수 있는 검색 결과 브라우징 부분으로 구성되어 있다. 질의 생성 부분은 궤적 기반 질의, 의미 기반 질의, 그리고 행위자(선수 이름) 기반 질의가 가능하도록 구성하였다. 검색 결과 브라우징 부분은 검색 결과의 수와 질의에 만족하는 축구 비디오 샷의 축구공의 궤적을 비트맵 이미지(*.bmp)로 저장한 아이콘을 출력한다. 사용자 질의 중에 궤적 기반 질의는 사용자 인터페이스의 왼쪽 상단에 있는 축구 경기장 모델 위를 사용자가 마우스 왼쪽 버튼을 클릭해서 축구공의 움직임을 표시하고 마지막 움직임에서 더블클릭 함으로써 원하는 축구공의 궤적을 포함하고 있는 축구 비디오 샷을 검색하기 위한 사용자 질의 궤적을 생성할 수 있다. 행위자 기반 질의는 [player]의 'select' 체크 박스를 클릭해서 원하는 행위자(선수 이름)을 선택할 수도 있거나 혹은 직접 텍스트 박스에 입력할 수도 있다. 마지막으로, 의미 기반 질의는 콤보 박스에서 원하는 의미 정보 중에 하나를 선택하면 된다. (그림 11)은 검색 GUI를 이용하여 질의를 생성한 예로 사용자가 직접 그린 축구공의 궤적을 가지면서, 궤적을 이루는 각각의 움직임 요소와 관련 있는 선수 이름으로 '김도근', '서정원', '안정환', '황선홍' 이며 질의 의미 정보로는 '오른쪽 센터링'을 나타내고 있다.



(그림 11) 이동 객체의 궤적 기반 축구 비디오 검색 GUI

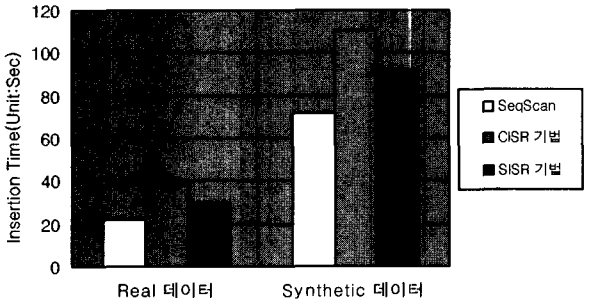
성능 평가를 위한 실험 데이타로는 real 데이타와 synthetic 데이타 두 가지를 사용한다. 먼저, real 데이타는 "골인되는(goal in) 장면"을 포함하고 있는 실제 축구 비디오 샷 350개로부터 축구공의 궤적을 추출한 것으로 대용량의 궤적 데이타를 생성하기 위해, 이를 변형 및 확장시켜 만든 460,000건의 축구공의 궤적 데이타이다. synthetic 데이타는 임의의 궤적 데이타를 생성할 수 있는 무작위 궤적 데이타 생성 프로그램을 이용하여 방향(각도), 거리 및 궤적을 구성하는 움직임 요소의 수를 다양하게 적용시켜 생성한 1,000,000건의 궤적 데이타이다. 기존의 유사 부분궤적 검색과 밀접한 관련이 있는 Shan과 같은 타 연구에서는 특별한 접근 기법을 사용하지 않고 순차적으로 궤적 데이타를 저장해서 검색하기 때문에, 본 논문에서는 성능 평가를 위해 궤적 데이타가 저장되어 있는 궤적 데이타 파일을 순차적으로 탐색하는 순차 탐색 방법(이하 SeqScan)과 제안하는 SISR 기법과 CISR 기법을 비교 수행한다. 또한 성능 평가는 삽입 시간, 검색 시간, 부가 저장공간으로 나누어 수행한다[15, 16]. <표 1>은 본 논문에서 사용하는 실험 인자를 나타낸다.

<표 1> 실험에 사용하는 실험 인자

데이터 종류 / 인자	Real 데이타	Synthetic 데이타
실험 데이타 수	460,000건	1,000,000건
움직임 요소의 수	1개 ~ 15개	
질의의 수	100개	
움직임 요소당 시그니처 비트 수	4 bit(CISR 기법), 64bit(SISR 기법)	
디스크 페이지 크기	8192B(8KB)	

삽입 시간은 실제 축구 비디오 샷으로부터 추출된 궤적 정보 즉, 움직임 요소가 발생한 위치 정보(location), 해당 위치에서 이동 객체(축구공)를 소유한 객체명(예 : 축구공을

소유한 선수이름), 이동 객체의 방향과 거리로 구성된 움직임 요소(motion)를 시그니쳐 파일과 궤적 데이타 파일로 구성된 인덱스 파일에 저장하는 데 소요되는 시간을 의미한다. (그림 12)는 Real 데이타와 Synthetic 데이타를 각각 삽입하는 데 소요되는 시간을 측정한 결과이다. SeqScan 방법은 특별한 접근 기법을 사용하지 않는 순차적인 저장 방식이기 때문에 이동 객체의 궤적 데이타를 단지 데이타 파일에만 저장하는 시간을 의미한다.

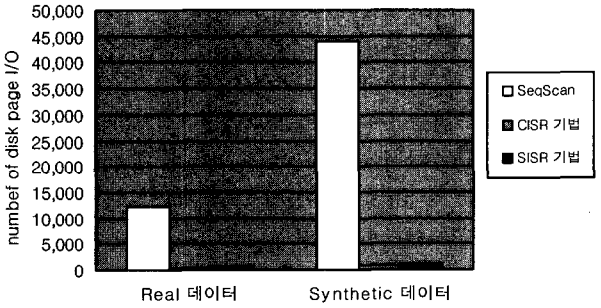


(그림 12) 데이타 삽입 시간

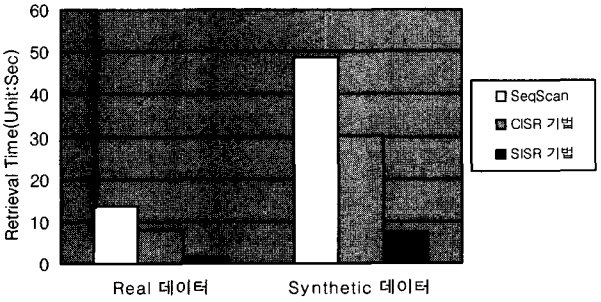
여기서, 세로축은 삽입 시간을 초 단위로 나타낸 것이며, 가로축은 해당 데이타 종류를 나타낸다. 삽입 시간은 메모리에서의 연산 시간(CPU 시간)과 디스크 접근 시간을 모두 포함한 전체 시간(Wall 시간)을 의미한다. 성능 평가 결과, real 데이타의 경우, SeqScan 방법, CISR 기법, SISR 기법에 대해서 각각 약 22초, 38초, 30초가 소요된다. synthetic 데이타의 경우, 각각 약 71초, 110초, 91초가 소요된다. 따라서, SeqScan 방법이 CISR 기법이나 SISR 기법에 비해 삽입 시간 성능이 좋다. CISR 기법은 시그니쳐 레코드의 구조적인 특성상 가변길이 레코드이므로 이를 관리하기 위한 별도의 링크 파일이 더 필요하므로 고정길이 시그니쳐 레코드 형식을 가지는 SISR 기법에 비해서 삽입 시간이 더 요구된다. CISR 기법이나 SISR 기법은 시그니쳐 파일의 특성상 SeqScan 방법에 비해 약 1.3배~1.7배 정도의 삽입 시간이 더 요구된다.

검색 성능을 평가하기 위해 real 데이타와 synthetic 데이타로부터 무작위로 각각 100개의 질의 데이타를 추출한 후, 이를 이용해서 10번의 질의 횟수를 통해 평균값을 구해서 검색에 요구되는 디스크 페이지 접근횟수(disk I/O)와 탐색 시간(search time)으로 나누어 성능을 측정한다. 여기서 SeqScan 방식의 경우는 궤적 데이타들이 저장된 데이타 파일만을 탐색하는 경우로 제한하며, 디스크로부터 메모리로 데이타를 읽어오는 경우 디스크 페이지 크기인 8192B (8KB)로 가정해서 읽어 들인다. 따라서 SeqScan 방식의 경우 전체 데이타 파일의 크기를 디스크 페이지 크기로 나누어서 구하면 하나의 질의에 대해서 필요한 디스크 접근 횟수를 구할 수 있다.

(그림 13)은 synthetic 데이타와 real 데이타에 대해서 각각 질의 100개에 대해서 10번의 질의 횟수를 통해 평균을 구한 디스크 접근 횟수를 나타낸다. real 데이타의 경우, SeqScan 방법은 약 12,000번, CISR 기법은 약 330번, 그리고 SISR 기법은 540번이다. synthetic 데이타의 경우는 약 43,000번, 약 910번, 약 1210번이다. 두 가지 데이타 모4두에 대해서 제안하는 CISR 기법과 SISR 기법이 SeqScan 방법에 비해 월등히 좋은 성능을 보인다. 이는 두 기법 모두 데이타 파일을 접근하기 전에 시그니쳐 파일을 통해서 많은 수의 불필요한 데이타를 제거하는 필터링 과정을 거치므로 실제로 데이타 파일을 접근해야 하는 디스크 접근 횟수가 매우 적어진다. 특히 디스크 접근 횟수 측면만을 고려한다면, CISR 기법이 SISR 기법에 비해서 좀 더 나은 성능을 나타낸다. 이는 시그니쳐 파일내의 궤적 시그니쳐들을 탐색해서 필터링하는 과정의 차이 때문이다. 즉, CISR 기법의 경우 질의 궤적 시그니쳐를 이용하여 궤적 시그니쳐들을 탐색해서 필터링할 때, 움직임 요소들 간의 순서를 고려해 처음 움직임 시그니쳐에서부터 마지막 움직임 시그니쳐까지 하나씩 옮겨가며 움직임 시그니쳐들 간의 거리의 차를 이용한다. 그에 반해, SISR 기법의 경우는 단순히 데이타 궤적 시그니쳐에 대해서 비트 단위 연산을 통해 질의 궤적 시그니쳐를 포함하는 것만을 후보 시그니쳐로 추출하기 때문이다.



(그림 13) 검색 시 요구되는 디스크 접근 횟수(I/O)



(그림 14) 검색 질의에 대한 응답 시간(CPU 시간 + I/O 시간)

(그림 14)는 질의에 대한 전체 응답 시간으로 메모리에서 계산하는 데 소요되는 CPU 시간과 시그니쳐 파일과 데이타 파일을 읽는 데 걸리는 디스크 접근 시간을 모두 합한 시간을 측정한 결과이다. real 데이타의 경우, SeqScan

방법은 약 13.5초, CISR 기법은 약 8.2초, 그리고 SISR 기법은 1.4초이며, synthetic 데이타의 경우, SeqScan 방법은 약 48.7초, CISR 기법은 약 30.2초, 그리고 SISR 기법은 약 7.4초가 소요된다. 결론적으로, CISR 기법이 SeqScan 방식에 비해 약 1.5배 정도의 검색 성능을 보이는 반면에, SISR 기법은 약 7배~10배 정도의 검색 성능을 보인다. 결론적으로 SeqScan 방법과 비교해서 제안하는 CISR 기법과 SISR 기법이 디스크 접근 횟수 측면보다는 검색 시간 측면에서 성능 차이가 줄어든다. 이는 SeqScan 방법의 경우 순차적으로 저장되어 있는 궤적 데이타 파일을 읽어 들이는 순차 탐색을 수행하는 데 반해, CISR 기법이나 SISR 기법의 경우 시그니쳐 파일부터 얻은 후보 시그니쳐들에 대해서만 그에 해당하는 궤적 데이타만을 궤적 데이타 파일부터 읽어 들이는 직접 탐색을 수행하기 때문이다. 즉, 직접 탐색의 경우, 디스크를 접근할 때마다 비용이 많이 요구되는 디스크 탐색 시간(seek time)이 필요하기 때문이다.

부가 저장 공간 비율(Storage Overhead : SO)은 식 (15)와 같이 원래의 궤적 데이타를 저장하는 데 필요한 저장 공간을 기준으로 인덱스 파일에서 부가적으로 더 요구되는 저장공간의 비율을 의미한다.

$$SO = \frac{\text{색인 파일의 크기}}{\text{데이타 파일의 크기}} \times 100 \qquad (15)$$

인덱스 파일에 각각 두 가지 데이타를 저장할 경우 부가 저장공간 비율은 <표 2>와 같다. SeqScan 방법의 경우는 순수한 궤적 데이타를 데이타 파일에 순차적으로 저장하는 방식이기 때문에 부가 저장공간이 필요하지 않으므로 부가 저장공간 비율은 0% 이다. real 데이타의 경우, CISR 기법과 SISR 기법은 각각 약 4%와 5% 정도의 부가 저장 공간을 더 요구한다. synthetic 데이타의 경우, 각각 약 3%와 2% 정도밖에 더 요구하지 않는다. 데이타베이스의 크기가 더 커질수록 시그니쳐 파일의 특성상 부가 저장 공간의 오버헤드에 부담이 줄어든다.

〈표 2〉 부가 저장 공간 비율

방법 \ 데이타	Real 데이타	Synthetic 데이타
SeqScan 방법	0 %	0 %
CISR 기법	5 %	4 %
SISR 기법	3 %	2 %

마지막으로, 비디오 데이터 내의 이동 객체의 궤적을 기반으로 제안되어진 기존의 연구들은 대부분이 궤적 정보를 모델링하고 주어진 질의 궤적과 데이터 궤적 사이의 유사성을 측정하기 위한 유사도 측정 기법에 대해서만 연구가 이루어졌다. 따라서 본 논문에서는 제안하는 방법을 데이터 모델링과 궤적 기반 검색 측면에서 기존의 연구들과 비교하면 다음 <표 3>과 같다.

〈표 3〉 기존의 연구들과의 비교

분 류	Shan의 기법	Li의 기법	Shepherd의 기법	Dagtas의 기법	제안하는 기법
궤적 데이터 모델링	방향 정보 (실제각도)	방향 정보 (코드화)	방향 정보 (코드화)	중심 좌표	방향(실제 각도), 거리 정보
궤적 기반 검색 지원	지원	지원	지원	지원안함	지원
유사성 측정 기법 지원	지원	지원	지원안함	지원	지원
색인 기법 지원	지원안함	지원안함	지원안함	지원안함	지원

5. 결론 및 향후 연구

본 논문에서는 이동 객체가 이루는 궤적에 대한 효율적인 저장 및 유사 부분궤적 검색을 지원하는 새로운 시그니쳐-기반 색인 기법을 제안하였다. 제안하는 시그니쳐-기반 색인 기법은 궤적 데이타를 토대로 궤적 시그니쳐를 생성하는 방법에 따라 합성 시그니쳐-기반 색인 기법과 중첩 시그니쳐-기반 색인 기법으로 나뉜다. CISR 기법은 시그니쳐 파일내의 궤적 시그니쳐 레코드가 가변 길이 레코드로 구성되며, 궤적 데이타를 구성하는 움직임 요소들의 수가 가변이라는 특징과 움직임 요소들의 순서 정보가 중요하다는 특징을 고려하여 이웃한 움직임 요소들 간의 각도의 차를 이용해서 움직임 시그니쳐를 생성하고 이들을 모두 합성(concatenated)시켜 궤적 시그니쳐를 구성한다. 그에 반해, SISR 기법은 시그니쳐 파일 내의 궤적 시그니쳐 레코드가 고정길이 레코드로 구성되며, 궤적을 구성하는 첫 번째 움직임 요소에서부터 시작하여 마지막 움직임 요소까지 순회하면서 이웃한 움직임 요소들 간의 각도를 이용하여 고정된 크기의 움직임 시그니쳐를 생성하고 이들을 모두 중첩(superimposed)시켜 궤적 시그니쳐를 구성한다. 두 가지 기법 모두 생성된 궤적 시그니쳐 정보는 시그니쳐 파일에 저장되고 검색시 주어진 사용자 질의 궤적 정보를 기반으로 데이타 파일을 직접 접근하기 전에 전체 궤적 시그니쳐들을 탐색하여 필터링을 수행함으로써 데이타 파일의 검색 범위를 현저히 줄임으로써 검색 성능을 향상시킨다. 또한 검색된 궤적 데이타와의 질의 궤적 데이타간의 유사성을 측정하기 위해 k-워핑 알고리즘을 적용시켜 검색의 효율성을 높인다. 아울러, 제안한 기법의 효율성을 측정하기 위해, 삽입 시간, 검색 시간, 부가 저장 공간을 토대로 Real 데이타와 Synthetic 데이타를 이용해서 순차 접근 방식(Seq- Scan), 제안하는 CISR 기법과 SISR 기법에 대해서 성능 비교를 수행하였다. 성능 평가 결과, 삽입 성능 측면에서는 Real 데이타의 경우, 제안하는 CISR 기법과 SISR 기법이 각각 SeqScan에 비해 약 1.7배와 1.3배 정도의 삽입 시간을 더 요구하며, 검색 성능 측면에서는 질의시 전체 응답 시간을 측정한 결과 제안하는 CISR 색인 기법이 SeqScan에 비해 두 가지 데이타 모두에 대해서 약 7~10

배 정도의 성능 향상을 보인다. 아울러 제안하는 기법은 시공간 데이터베이스, 모바일 데이터베이스, 비디오 데이터베이스 등과 같은 응용 분야에서 이동 객체가 이루는 궤적을 기반으로 하는 사용자의 질의 처리시에 우수한 검색 성능을 보장할 수는 색인 기법으로 사료된다.

앞으로의 연구로는 제안하는 CISR 색인 기법과 SISR 색인 기법 모두 구조적인 특성상 시그니쳐 파일 전체를 순차적으로 탐색해야 하는 오버헤드를 가지고 있다. 따라서 검색 성능을 향상시키기 위해, 멀티 쓰레드 프로그래밍(multi-thread programming) 기법을 이용하여 다수의 디스크에 동시에 나누어서 저장하고 검색할 수 있는 병렬 처리(parallel processing) 기법을 연구하는 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 M. Stonebraker, "Chabot : Retrieval from a Relational Database of images," IEEE Computer, Vol.28, No.9, pp.40-48, 1995.

[5] 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.

[6] C. Faloutsos and S. Christodoulakis, "Signature files : An access methods for documents and its anylytical performance evaluation," ACM Transaction on Database Systems, Vol.2, No.4, pp.267-288, 1984.

[7] 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.

[8] M. K. Shan and S. Y. Lee, "Content-based Video Retrieval via Motion Trajectories," In Proc. International Conference on SPIE Electronic Imaging and Multimedia System II, pp. 52-61, 1998.

[9] J. Z. Li, M. T. Ozsu and D. Szafron, "Modeling Video Temporal Relationships in an Object Database Management System," In Proc. of Multimedia Computing and Networking(MMCN97), pp.80-91, 1997.

[10] M. Nabil, A. H. Ngu and J. Shepherd, "Modeling Moving Objects in Multimedia Databases," In Proc. of 5th International Conference on Database Systems for Advanced Applications, pp.67-76, 1997.

[11] S. Dagtas, A. Ghafoor and R. L. Kashyap, "Motion-based Indexing and Retrieval of Video using Object Trajectories," In Proc. of 6th Workshop on Multimedia Information Systems, pp.33-41, 2000.

[12] B. K. Yi, H. V. lagadish and C. Faloutsos, "Efficient Retrieval of Similar Time Sequences Under Time Warping," In Proc. International Conference on Data Engineering, pp. 201-208, 1998.

[13] S. H. Park, et al., "Efficient Searches for Similar Subsequence of Difference Lengths in Sequence Databases," In Proc. International Conference on Data Engineering, pp. 23-32, 2000.

[14] 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. International Conference on Data Engineering, pp.607-614, 2001.

[15] 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.

[16] G. Salton and M. McGill, An introduction to Modern Information Retrieval, McGraw-Hill, 1993.

심 춘 보

e-mail : cbsim@cup.ac.kr

1996년 전북대학교 컴퓨터공학과(공학사)

1998년 전북대학교 컴퓨터공학과(공학석사)

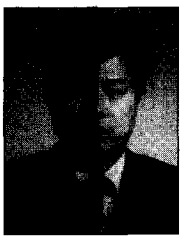
2003년 전북대학교 컴퓨터공학과(공학박사)

1996년~1997년 한국전자통신연구원 위촉연구원

2003년~2004년 한국과학재단 신진연구자 연수지원사업 연구원

2004년~현재 부산가톨릭대학교 컴퓨터정보공학부 전임강사

관심분야 : 멀티미디어 데이터베이스, 멀티미디어 정보검색, 시공간 데이터베이스, LBS 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 교수

관심분야 : 멀티미디어 데이타베이스, 멀티미디어 정보검색, 고차원 색인 기법, 하부저장구조 등