의미 정보를 이용한 다차원 데이터 시퀀스의 유사성 척도 연구

이 석 룡[†] · 이 주 홍[††] · 전 석 주[†††]

요 약

연속된 일 차원 실수로 이루어진 시계열 데이터는 데이터 마이닝이나 데이터 웨어하우징과 같은 다양한 데이터베이스 응용 분야에서 연구되어져 왔다. 그러나 최근의 복잡한 비즈니스 환경에서, 다차원 데이터 시퀀스(multidimensional data sequence : MDS)는 일 차원 시계열 데이터와 더불어 그 중요성이 더해가고 있다. 다차원 데이터 시퀀스의 예로써, 비디오 스트림은 색상과 질감 등의 속성들로 이루어진 다차원 공간상에서 MDS로 나타낼 수 있다. 본 논문에서는 패턴 유사성 검색에서 사용되는 효과적인 유사성 척도를 제시한다. 하나의 MDS는 여러 개의 세그먼트(segment)로 나누어지며, 각 세그먼트는 다양한 의미적인 특징들로 표현된다. 유사성 척도는 이러한 세그먼트에 대해서 정의되는데 이 척도를 사용하여 어떤 주어진 질의 시퀀스에 대하여 무관한 세그먼트들은 검색 대상에서 일차적으로 제외된다. 데이터 시퀀스와 질의 시퀀스 모두 세그먼트 단위로 분할되며, 질의 처리는 전체 시퀀스의 모든 데이터를 검색하지 않고 데이터 세그먼트와 질의 세그먼트의 특징을 비교하는 것을 기초로 하여 수행된다.

A Study of Similarity Measures on Multidimensional Data Sequences Using Semantic Information

Seok-Lyong Lee[†] · Ju-Hong Lee[††] · Seok-Ju Chun[†††]

ABSTRACT

One-dimensional time-series data have been studied in various database applications such as data mining and data warehousing. However, in the current complex business environment, multidimensional data sequences (MDS') become increasingly important in addition to one-dimensional time-series data. For example, a video stream can be modeled as an MDS in the multidimensional space with respect to color and texture attributes. In this paper, we propose the effective similarity measures on which the similar pattern retrieval is based. An MDS is partitioned into segments, each of which is represented by various geometric and semantic features. The similarity measures are defined on the basis of these segments. Using the measures, irrelevant segments are pruned from a database with respect to a given query. Both data sequences and query sequences are partitioned into segments, and the query processing is based upon the comparison of the features between data and query segments, instead of scanning all data elements of entire sequences.

키워드 : 다차원 데이터 시퀀스(Multidimensional Data Sequence), 유사성 척도(Similarity Measure), 패턴 검색(Pattern Retrieval)

1. 서 론

시계열 데이터는 데이터 마이닝이나 데이터 웨어하우징과 같은 다양한 데이터베이스 응용 분야에서 중요하게 다루어져 왔다. 이는 주가나 상품의 가격, 날씨의 패턴, 판매 지수, 생리적 측정 데이터과 같이 시간적 변화에 따라 나타나는 실수로 된 하나의 시퀀스를 가지고 있다. 그러나 시계열 데이터는 일 차원 데이터의 연속이어서 지금까지의 연구는 데이터의 일 차원 시퀀스를 인덱싱하거나 검색하는 데에 초점을 두었다. 오늘날 비즈니스 환경이 복잡해지고 다차원 데이터가 많은 응용 프로그램 영역에서 폭 넓게 사용됨에 따라 다차원 데이터를 효과적으로 검색하는 것이 중요해지고 있다. 최근의 연구[12]에서 우리는 n-차원 공간에서 K 개로 구성된 MDS S를 벡터 요소의 연속으로 다음과 같이 정형화하여 정의하였다. : $S = S[1], S[2], \cdots, S[K]$. 여기에서 각 요소 벡터 $S[j](1 \le j \le K)$는 n개의 스칼라 양으로 구성되어지며, $S[j] = (S^1[j], S^2[j], \cdots, S^n[j])$으로 표현된다. 위의 정의에서 시계열 데이터는 시퀀스 내의 각 데이터 요소들을 하나의 스칼라 값으로 치환함으로써 모델링 할 수 있다. MDS의 전형적인 예로써, 비디오 스트림은 다수의 프레임들로 구성되며 각 프레임은 색상, 질감, 형태 등과 같은 여러 가지 특징 속성(feature attribute)들로 나타

※ 본 연구는 2002년 정보통신연구진흥원 산업기술개발사업에 의하여 지원되었음.
† 정 회 원 : 한국외국어대학교 산업정보시스템공학부 교수
†† 종신회원 : 인하대학교 컴퓨터공학부 교수
††† 정 회 원 : 안산1대학 인터넷정보과 교수
논문접수 : 2002년 11월 5일, 심사완료 : 2003년 2월 13일

낼 수 있다. 예를 들면, 프레임은 프레임 전체나 프레임 내의 분할된 블록 안의 색상 픽셀의 평균 값을 사용하여 RGB 혹은 YCbCr 색상 공간에서의 다차원 벡터로서 표현될 수 있다. 비디오 스트림은 다차원 데이터 공간에서 점들의 궤적으로 모델링될 수 있는 데, 각 점은 다차원 벡터로 나타낼 수 있으며 이 벡터의 요소는 각 프레임들의 특징 값들을 나타낸다.

다차원 데이터 시퀀스는 여러 개의 세그먼트(segment)들로 분할되며, 유사성 척도는 이러한 세그먼트들을 기초로 정의되어진다. MDS를 분할하기 위해서는 [11]에서 제시했던 분할 기술을 활용한다. 본 논문에서는 다차원 시퀀스 데이터베이스에서 유사성에 근거한 패턴 검색을 위한 효과적인 유사성 척도들을 제시한다. 이 척도들은 다차원 공간에서 세그먼트 간의 거리와 세그먼트의 의미적(semantic) 측면을 고려하여 정의되어진다.

첫째, 데이터베이스로부터 질의와 무관한 세그먼트들을 제거하기위해 'no false dismissal'을 보장하는 거리 함수를 정의한다. 다음으로, 세그먼트 내의 점들이 움직이는 방향과 같은 세그먼트의 방향적 특징들, 세그먼트의 볼륨(volume)이나 에지(edge)와 같은 기하학적 특징을 고려한 의미적 척도를 제시한다. 한편, 세그먼트 형태로써 하이퍼 사각형(hyper-rectangle)을 채택했으며, 이는 R-tree[6]나 R-tree의 변형된 형태의 인덱싱 방법들[2, 3, 15]과 같은 현재 지배적인 인덱싱 메커니즘들이 노드의 형태를 나타내는 데에 최소 경계 사각형(minimum bounding rectangle : MBR)을 사용하고 있고, 따라서 이 인덱싱 메커니즘들을 변형시키지 않거나 약간의 변형 만으로도 활용할 수 있기 때문이다. 데이터 시퀀스와 질의 시퀀스는 모두 세그먼트 단위로 분할되어지며, 질의 처리는 전체 시퀀스들의 모든 점 단위로 검색하는 것이 아니라 하이퍼사각형 형태의 세그먼트들을 기초로 수행됨으로써 검색 성능을 향상시킨다.

본 논문의 나머지 부분은 다음과 같은 구성으로 이루어진다. 2장에서는 시퀀스 데이터의 유사성 척도들과 유사성 검색 방법에 관한 관련 연구들을 간단히 소개한다. 3장에서 다차원 데이터 시퀀스의 분할 방법을 설명하고, 유사성 척도들에 대해서는 4장에서 제시한다. 5장에서는 비디오 데이터와 가상으로 생성된 시퀀스들에 대해 수행된 실험 결과를 보여 주며, 6장에서 결론을 기술한다.

2. 관련 연구

시계열 데이터에 관한 다양한 유사성 검색 방법이 제안되었으며, 각각의 기법은 나름대로의 유사성 척도들을 채택하여 검색을 수행하고 있다. Agrawal et al.[1]이 연구한 전체 시퀀스 매칭(whole sequence matching) 방법에서는 비교되는 두 시퀀스의 길이는 같아야 한다. 그들은 시간에 관한 시퀀스를 주파수 도메인의 값으로 변환하고, 고차원 문제를 해결하기 위해서 이산 푸리에 변환(Discrete Fourier Transform : DFT)을 사용하였다. 이 기법에서는 두 시퀀스 사이의 유사성을 측정하기 위하여 변환된 주파수 도메인에서 유클리디안(Euclidean) 거리를 사용했다.

Faloutsos et al.[4]은 서로 다른 길이의 시퀀스들 사이의 유사성 검색을 고려한 빠른 서브 시퀀스 매칭(fast sub-sequence matching) 기법을 제시했다. 그들은 데이터 시퀀스에 대하여 길이가 w인 슬라이딩 윈도우를 사용하여 각 윈도우에 포함된 w개의 일 차원 값들을 w차원의 한 점으로 나타내고 DFT를 사용하여 이 점들을 저차원의 점들로 나타내었다. 저차원으로 변환된 데이터 시퀀스는 서브 시퀀스들로 나뉘는데, 각각은 MBR로 표현되고 *'ST-index'*를 사용하여 색인되고 저장된다. 그들은 데이터베이스로부터 질의와 무관한 MBR을 제거하기위해 질의 포인트와 MBR사이에 거리에 기반한 유사성 척도를 고안하였다. 그러나, 이 유사성 함수는 MBR 내부의 점들의 여러 가지 특성은 고려하지 않고 단지 Euclidean 거리에만 바탕을 두었다.

Rafiei et al.[14]은 주어진 시퀀스의 안전한 선형 변환들의 집합을 제안하였는데, 이것들은 시계열 데이터에 대한 유사성 질의의 기본으로 사용될 수 있다. 그들은 이동 평균(moving average), 역전(reversing), 타임 워핑(time warping)과 같은 함수들을 정형화하여 표현하였다. 이들 변환 함수들을 [13]에서 다중 변환(multiple transformation)으로 확장되었는 데, 각 함수들이 각각 따로 적용되지 않고 여러 변환들이 색인을 검색하는 데 동시적으로 적용되므로 색인을 여러 번 검색하지 않고 한번만 검색해도 된다는 이점이 있다.

최근의 연구로써, Yi et al.[16]에서는 임의의 L_P norm을 사용해 거리에 기반한 유사성 척도를 제시했다. 그리고 Keogh et al.[10]은 시퀀스의 압축을 위하여 APCA(Adaptive Piecewise Constant Approximation) 표현 방법을 제안하였으며, APCA를 기반으로 하는 인덱스 공간에서 검색을 위한 두 가지의 유사성 척도를 제시했는데, 'no false dismissal'을 보장하는 하한 유클리디안 (low-bounding Euclidean) 함수 D_{LB}와, 보장하지 않는 비하한 유클리디안 (non low-bounding Enclidean) 함수 D_{AE}이다.

앞에서 소개된 모든 방법들은 일 차원 시계열 데이터에 대한 유사성 검색을 다루고 있기 때문에 다차원 데이터 시퀀스에 적용될 수 없다. 또한, 위의 방법에서 제시한 유사성 척도들은 대부분 Euclidean 거리에 바탕을 두었기 때문에 시퀀스들의 기하학적, 방향적 특징들과 같은 의미적 측면을 고려하지 않았다. 또 한편으로는, [4, 7, 8, 17]과 같은 여러 가지 방법이 멀티미디어 검색 분야에서 제안되었으나, 이 방법들에서 사용된 유사성 척도들은 색상, 질감, 모양 등과 같은 멀티미디어 자료에 고유한 특성들을 다루는 것에 중점을 두고 있으며, 따라서 일반적인 다차원 데이터 시퀀스에는 적용하기가 어렵다.

3. 세그멘테이션 기법의 개요

본 논문에서 제안한 유사성 척도들은 앞에서 언급했듯이 세그먼트를 기초로 하여 정의되어진다. 유사성 척도들에 대하여 언급하기 전에, 먼저 세그멘테이션 기법[11]을 간단히 서술한다.

정의 1 세그먼트(Segment) : n차원공간에서 k개의 점 P_j $(j=1, 2, \cdots, k)$를 갖는 하이퍼사각형 형태의 세그먼트 SEG는 다음과 같이 두 개의 끝점 L(low point)과 H(high point), 그리고 사각형내의 점 수로 표현된다. : $SEG = <L, H, k>$. 여기에서 $L=\{(L^1, L^2, \cdots, L^n) \mid L^i = \min_{1\le j\le k}(P_j^i)\}$, $H = (H^1, H^2, \cdots, H^n) \mid H^i = \max_{1\le j\le k}(P_j^i) (i=1, 2, \cdots, n)$ 이다. ■

그러면, 세그먼트 SEG의 볼륨 $Vol(SEG)$와 에지 $Edge(SEG)$는 다음의 식으로 계산된다.

$$Vol(SEG) = \prod_{1\le i\le n}(SEG.H^i - SEG.L^i)$$

$$Edge(SEG) = 2^{n-1} \cdot \sum_{1\le i\le n}(SEG.H^i - SEG.L^i)$$

세그먼트의 특성을 측정하기 위한 정량적 척도로써, 다음 2가지의 파라미터를 사용한다 : volume per point(VPP), edge per point(EPP). MDS S가 p개의 세그먼트 $SEG_1, \cdots, SEG_p$로 분할된다 하면, 시퀀스 S의 VPP와 EPP는 다음 식으로 계산된다.

$$VPP = \frac{\sum_{1\le j\le p} Vol(SEG_j)}{\sum_{1\le j\le p} SEG_j.k}, \quad EPP = \frac{\sum_{1\le j\le p} Edge(SEG_j)}{\sum_{1\le j\le p} SEG_j.k}$$

세그멘테이션은 사전에 정의된 조건을 만족했을 때 시퀀스 상의 포인트를 세그먼트로 병합시키는 반복적인 절차이다. 시퀀스 S의 한 점 P가 n 차원의 공간 $[0, 1]^n$ 상에서 세그먼트에 병합되는 과정을 생각해 보자. 세그멘테이션은 다음과 같은 과정을 거쳐서 일어난다. : 시퀀스 S의 한 점 P를 세그먼트 SEG에 병합할 때 주어진 조건을 만족시키면 현재의 세그먼트 SEG에 병합하고, 만족시키지 못할 경우 그 점으로부터 새로운 세그먼트가 시작되도록 하여 계속 이 과정을 반복한다. 공간이 한정된 관계로 상세한 세그멘테이션 알고리즘은 언급하지 않는다. 이것에 관심이 있다면 [11]을 참조하면 세그멘테이션 조건들과 알고리즘, 그리고 실험적 결과들에 대한 자세한 내용을 볼 수 있다.

4. 유사성 척도(Similarity measures)

4.1 거리에 기반한 유사성 척도

다차원 공간 상에서 다차원 벡터로 표현되는 두 객체간의 유사성은 일반적으로 두 점간의 유클리디안 거리(Euclidean distance)의 함수로써 정의된다. 세그먼트간의 유사성도 각각의 특징 벡터 사이의 거리의 함수로 정의될 수 있다. 두 객체 간의 거리는 $[0, \infty]$ 사이의 값을 갖는 데 비하여 유사성의 값의 범위는 $[0, 1]$이다. 두 객체가 유사하면 거리는 가까워지고 다르면 멀어지게 된다. 이와 반대로, 유사성의 값은 두 객체가 유사하면 1에 가깝고 다르면 0에 가까워 진다. 두 객체 사의의 거리는 적절한 사상(mapping) 함수를 사용하여 유사성으로 쉽게 변환될 수 있다. 본 논문에서는 데이터 공간이 각 차원의 길이가 1인 $[0, 1]^n$의 하이퍼 큐브로 정규화되어 있다고 가정한다. 따라서, 이 큐브에서 두 점간의 최대 거리는 대각선의 길이인 $\sqrt{n}$이 된다. 이 성질을 이용하여 거리를 유사성으로 쉽게 사상할 수 있으며, 표현을 단순화하기 위하여 본 논문에서는 유사성의 척도로써 거리를 사용한다.

4.1.1 시퀀스 사이의 거리

두 개의 다차원 데이터 시퀀스 S_1과 S_2를 고려하자. n차원 시퀀스 S_1과 S_2에 각각 포함되어 있는 두 개의 임의의 점들 사이의 거리는 다음의 식으로 주어진다.

$$d(S_1[i], S_2[j]) = \left(\sum_{1\le t\le n} |S_1[i, t] - S_2[j, t]|^2\right)^{1/2}$$

여기에서 $S_1[i]$와 $S_2[j]$는 각각 시퀀스 S_1과 S_2의 i-번째와 j-번째 점을 나타내며, t는 1부터 n까지의 차원을 나타낸다.

시퀀스간의 거리의 의미는 점들간의 거리의 의미와는 다르다. 다차원 데이터 시퀀스는 여러 개의 다차원 점들로 구성된다. 두 시퀀스의 거리로서 두 시퀀스 안의 점들의 쌍의 거리의 합을 사용하는 것은 적절하지 않다. 그 이유는 서로 유사한 두 시퀀스의 점의 개수가 많으면 그들의 거리 합의 값은 유사하지 않지만 점의 개수가 적은 두 시퀀스의 거리보다 더 클 수도 있다. 본 논문에서 사용할 두 시퀀스 사이의 거리를 제시한다. 먼저, 길이가 같은 두 시퀀스 사이의 거리를 정의하고, 이것을 길이가 다른 시퀀스 간의 거리를 구하는 식으로 확장한다.

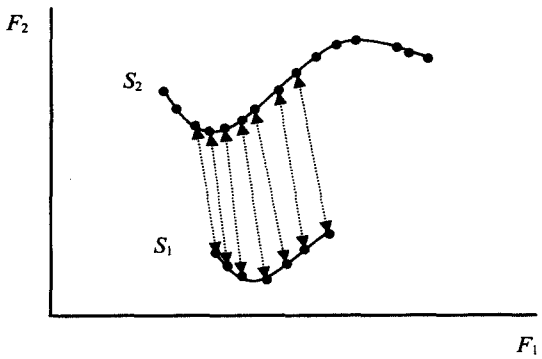
정의 2 : 각각 k개의 점들을 가지고 있는 서로 길이가 같은 다차원 시퀀스 S_1과 S_2의 거리 $D(S_1, S_2)$는 다음과 같이 두 시퀀스 내의 각 점 사이의 평균 거리로 정의된다.

$$D_s(S_1, S_2) = D_{mean}(S_1, S_2) = \frac{1}{k} \cdot \sum_{i=0}^{k-1} d(S_1[i], S_2[i]) \quad ■$$

다음으로 서로 길이가 달라서 점과 점들을 일 대 일로 대응시킬 수 없는 경우를 생각하자. 이 경우에는 짧은 시퀀스를 긴 시퀀스의 앞 부분에서 끝부분까지 슬라이딩하면서 대응시켜 거리를 계산하고 이중에서 가장 짧은 거리를 두 시퀀스 사이의 거리로서 정의한다. 좀 더 형식적으로 정의하면 다음 정의와 같다.

정의 3 : 길이가 서로 다른 두 다차원 시퀀스 S_1과 S_2간의 거리 $D(S_1, S_2)$는 각 시퀀스가 각각 p개와 q개의 점을 포함한다고 할 때 ($p \le q$), S_1과 S_2의 가능한 모든 서브 시퀀스간의 최소 평균 거리로 정의되며, 시퀀스 S의 a번째 점부터 b번째 점으로 이루어진 서브 시퀀스를 $[a:b]$라 표기할 때, 다음과 같이 표현된다.

$$D_s(S_1, S_2) = \min_{1 \le j \le q-p+1} D_{mean}(S_1[1:p], S_2[j:j+p-1]) \blacksquare$$



(그림 1) 길이가 서로 다른 두 다차원 시퀀스 S_1과 S_2간의 거리

(그림 1)은 거리 D_S를 도식적으로 보여 준다. 시퀀스 S_1의 연속된 점들이 시퀀스 S_2 내의 같은 수의 연속된 점들과 시퀀스 S_2의 시작점부터 시작하여 끝점까지 비교된다. 이 과정에서 계산된 거리의 값들 중에서 가장 작은 거리가 D_S가 된다.

4.1.2 세그먼트 사이의 거리

두 세그먼트 사이의 거리 D_{seg}를 측정하기 위하여 다차원 공간 상에서 두 세그먼트의 상대적 위치를 고려하여 다음과 같이 정의한다.

정의 4 : n 차원의 유클리디안 공간에서 두 개의 세그먼트 SEG_1과 SEG_2간의 거리 D_{seg}는 각 세그먼트 내의 점들을 최소 공간으로 모두 포함하는(minimum-bounding) 두 하이퍼 사각형 사이의 최소 거리로서 정의되며, 다음과 같이 표현된다.

$$D_{seg}(SEG_1, SEG_2) = \sqrt{\sum_{i=1}^{n} p_i^2}$$

$$\text{where} \quad p_i = \begin{cases} |SEG_1.H_i - SEG_2.L_i| & \text{if } SEG_1.H_i < SEG_2.L_i \\ |SEG_1.L_i - SEG_2.H_i| & \text{if } SEG_2.H_i < SEG_1.L_i \\ 0 & \text{otherwise} \end{cases}$$

■

관찰 5 : 두 개의 세그먼트 SEG_1과 SEG_2가 점 P_1, P_2를 각각 포함하고 있을 때, 두 세그먼트간의 거리 D_{seg}은 각 세그먼트에 포함되어 있는 두 점간의 어떤 쌍의 거리 보다 작다. 즉,

$$D_{seg}(SEG_1, SEG_2) \le \min_{P_1 \in SEG_1, P_2 \in SEG_2} d(P_1, P_2)$$

■

위의 관찰에 근거하여 D_{seg}와 D_S에 관하여 하한 관계(lower bounding relationship)를 보여주는 다음 정리가 성립한다. 여기에서, D_S는 두 개의 세그먼트 SEG_q와 SEG_t에 각각 포함되어 있는 두 시퀀스 S_q와 S_t 사이의 거리이다.

정리 6 (하한 거리(*Lower Bounding Distance*) : $D_{seg}(SEG_q, SEG_t) \le D_S(S_q, S_t)$) : 질의 시퀀스 SEG_q와 데이터 시퀀스 SEG_t사이의 거리 $D_{seg}(SEG_q, SEG_t)$는 두 세그먼트 SEG_q와 SEG_t에 각각 포함되어 있는 두 시퀀스 S_q과 S_t사이의 거리 $D_S(S_q, S_t)$에 하한(lower bounding)이다. 즉 :

$$D_{seg}(SEG_q, SEG_t) \le D_S(S_q, S_t)$$

증명 : 두 세그먼트 SEG_q와 SEG_t가 각각 k, l개의 점을 가지고 있고 ($k \le l$), P_q와 P_t가 각각 SEG_q, SEG_t에 포함된 임의의 점이라 하면 다음이 성립한다.

$$D_s(S_q, S_t) = \min_{1 \le j \le l-k+1} d_{mean}(S_q[1:k], s_t[j:j+k-1]) \ge \min_{P_q \in SEG_q, P_t \in SEG_t} d(P_q, P_t)$$

관찰 5에 의하여, 다음 식을 유도할 수 있다.

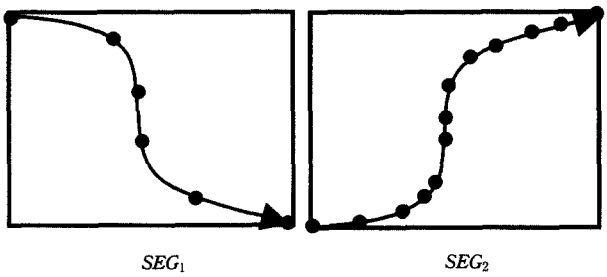
$$D_{seg}(SEG_q, SEG_t) \le \min_{P_q \in SEG_q, P_t \in SEG_t} d(P_q, P_t)$$

따라서, $D_{seg}(SEG_q, SEG_t) \le D_S(S_q, S_t)$라고 결론 지을 수 있다. ■

정리 6을 적용하여, *'false dismissal'*없이 데이터베이스로부터 질의에 무관한 세그먼트를 제거하기 위해 거리 함수 $D_{seg}(SEG_q, SEG_t)$를 사용할 수 있으며, 이는 이 거리 함수가 각각의 세그먼트에 포함된 두 시퀀스간의 거리 $D_S(S_q, S_t)$에 대하여 하한을 제공하기 때문이다.

4.2 의미적 척도(Semantic measures)

거리 함수 $D_{seg}(SEG_q, SEG_t)$가 비록 *'no false dismissal'*을 보장하고 있지만 D_{seg}가 가지고 있는 문제점은 검색의 효율에 중대한 영향을 미치는 수많은 '과오 선택(*false hits*)'을 초래한다는 점이다. 과오 선택이 증가할수록 과오 선택된 세그먼트들에 대하여 세그먼트 내의 모든 점들을 비교하는 비용이 많이 드는 사후 처리 과정(순차 탐색)을 거쳐야 하므로 필연적으로 효율이 저하된다. 또 다른 문제로써, 일반적으로 거리 함수는 세그먼트가 가지고 있는 의미적 측면을 고려하지 못한다는 점이다. 예를 들어, (그림 2)에서 보인 두 세그먼트를 고려해 보자.



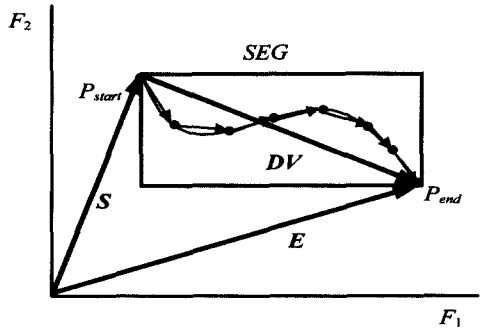
(그림 2) 두 세그먼트 사이의 의미적 차이

거리 함수는 각 세그먼트를 구성하고 있는 두 개의 하이퍼사각형 간의 거리만을 고려할 뿐이다. 그러나 (그림 2)에서 보인 것처럼 두 세그먼트는 거리 뿐만 아니라 여러 가지 의미적 요소들을 내포하고 있다. 첫째로, 각 세그먼트 안에 있는 점들은 움직이는 방향성을 가지고 있으며, 두 세그먼트는 이 방향성에서 분명히 차이가 있다. 움직임의 방향(moving direction)은 시퀀스의 패턴의 특성을 결정하는 데 중요한 요소이다. 다음으로, SEG_2는 SEG_1보다 내부의 점들이 조밀하다. 즉, SEG_2에는 SEG_1보다 점들이 더 많다. 반면에 두 세그먼트의 형태는 매우 비슷하다. 이러한 점들이 두 세그먼트의 유사성을 좀 더 정교한 방법으로 비교하는 의미적 척도를 연구하게 된 동기이다. 두 세그먼트의 유사성을 측정하기 위하여 세그먼트로부터 두 가지의 특성(방향과 기하학적 특성)을 추출한다.

4.2.1 방향 특성(Directional feature)

세그먼트는 방향적인 특성 (방향성)을 가지고 있다. 세그먼트안의 점들은 시간의 경과에 따라 시작점 P_{start}으로부터 끝점 P_{end}까지 움직이고 있다. (그림 3)은 세그먼트 내의 시작점과 끝점을 보여주고 있다. 주어진 세그먼트 SEG에서 방향 벡터(directional vector : $SEG.DV$)는 세그먼트의 시작점으로부터 끝점으로의 방향을 나타내는 벡터로써 정의된다. S가 다차원 공간의 원점으로부터 P_{start}로 향하는 벡터이고, E는 원점으로부터 P_{end}로 향하는 벡터라고 할 때, 방향 벡터 $SEG.DV$는 다음과 같이 표현된다.

$$SEG.DV = E - S$$



(그림 3) 세그먼트의 방향 벡터

두 세그먼트 SEG_q와 SEG_t의 방향 유사성(directional similarity : sim_D)은 각 세그먼트의 방향 벡터의 코사인을 취한 것으로 정의된다. 이것은 두 벡터의 유사성을 측정하기 위하여 많이 사용되는 방법으로써 코사인 유사성(*cosine similarity*)이라 불린다. 두 세그먼트의 방향 벡터 $SEG_q.DV$와 $SEG_t.DV$의 사이 각을 θ라 하면 $\cos\theta$는 두 벡터의 내적 (inner product)을 사용하여 다음과 같이 표시된다.

$$\cos\theta = \frac{SEG_q.\,DV \cdot SEG_t.\,DV}{|SEG_q.\,DV||\,SEG_t.\,DV|}$$

$\cos\theta$가 가질 수 있는 값의 범위가 [-1, 1]이므로 유사성을 표현하기 위하여 [0, 1] 범위 안의 값을 가지도록 정규화하여 표현할 수 있다. 이것은 다음 식을 사용하여 쉽게 변환할 수 있다.

$$sim_D(SEG_q, SEG_t) = \frac{1+\cos\theta}{2}$$

4.2.2 기하학적 특성(Geometric feature)

앞의 절에서 언급한 바와 같이 유사성은 일반적으로 0과 1사이의 값으로 표현된다. 임의의 변수 v에 대하여 두 객체 o_A와 o_B의 특성이 정량적인 값 v_A와 v_B로 표현된다고 할 때, 두 객체 사이의 상이도(difference) $diff_v(o_A, o_B)$를 고려해 보자. 변수 v에 대하여 두 객체 o_A와 o_B의 상이도는 단순하게 $|v_A - v_B|$과 같은 방법으로 계량할 수 있다. 그러나, 이 상이도 값의 범위는 변수 v의 값의 크기에 따라 많은 편차를 보인다. 따라서, 이 상이도 값의 범위가 0과 1사이가 되도록 정규화할 필요가 있다. 변수 v가 가질 수 있는 값의 범위, 즉 변수 v의 도메인을 $dom(v)$라 하고, 이 도메인의 상한과 하한을 $max(dom(v))$과 $min(dom(v))$이라 하자. 그러면 두 객체 o_A와 o_B의 상이도 $diff_v(o_A, o_B)$는 다음과 같이 나타낼 수 있다.

$$diff_v(o_A, o_B) = \frac{|v_A - v_B|}{\max(dom(v)) - \min(dom(v))}$$

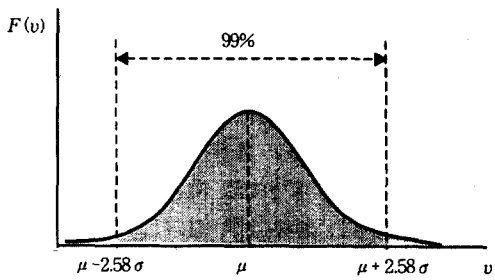
따라서, 두 객체 o_A와 o_B의 유사도 $sim_v(o_A, o_B)$는 범위 0과 1사이의 값을 갖도록 다음 식으로 표현할 수 있다.

$$sim_v(o_A, o_B) = 1 - diff_v(o_A, o_B)$$

두 세그먼트 SEG_q와 SEG_t간의 기하학적 유사도 sim_G를 VPP와 EPP를 사용하여 표현할 수 있다. VPP와 EPP는 세그먼트의 기하학적 특성을 정량적으로 나타내므로 다차원 시퀀스의 세그멘테이션 과정에서 중요하게 사용된다. sim_G를 결정하기 위해서는 먼저 VPP와 EPP값의 상한 값과 하한 값인 $max(dom(VPP))$, $min(dom(VPP))$, $max(dom(EPP))$, 그리고 $min(dom(EPP))$를 결정해야 한다.

변수 v의 어떤 시점에서의 값이 예외적으로 아주 큰 값

혹은 아주 작은 값을 가진다면 이 변수의 상한과 하한은 그 값에 크게 영향을 받고, 따라서 아주 큰 $max(dom(v))$ 값이나 혹은 아주 작은 $min(dom(v))$ 값을 갖게 된다. 이것은 $diff_v$를 아주 작게 만들어서 결국은 기대한 것 보다 매우 큰 sim_v값을 도출하므로 명백히 바람직한 현상이 아니다. 다시 말하면 두 객체 o_A와 o_B가 변수 v에 대하여 실제로는 서로 다르더라도 큰 유사도 값을 갖게 되어 서로 유사한 객체로 간주되는 것이다. 이러한 현상을 피하기 위하여 $max(dom(v))$과 $min(dom(v))$을 결정하는 과정에서 비정상적으로 큰 값과 작은 값을 배제하기 위하여 통계적 기법을 사용한다. 먼저 변수 v에 대하여 고정된 수의 표본 값을 임의로 추출하고, 이들의 평균 값 μ_v와 표준편차 σ_v를 계산한다. 변수 v값의 분포가 가우시안 분포(Gaussian distribution : $N(\mu_v, \sigma_v^2)$)를 따른다고 가정하면 대부분의 값들이 포함되게 되는 변수 v값의 범위를 계산할 수 있다. (그림 4)는 변수 v에 대하여 가우시안 분포의 확률 밀도 곡선(probability density curve)을 나타낸다.



(그림 4) 가우시안 분포의 확률 밀도 곡선

가우시안 분포 표를 참조하여[9], 대략 99%의 값들을 포함하는 범위를 쉽게 발견할 수 있다. 이 범위는 $P[v \le \mu_v + 2.58\sigma_v] = 0.9951$이므로 $[\mu_v - 2.58\sigma_v \le v \le \mu_v + 2.58\sigma_v]$가 된다. 즉, 변수 v의 최대값과 최소값으로써 $max(dom(v))$과 $min(dom(v))$을 취하지 않고, $\mu_v + 2.58\sigma_v$과 $\mu_v - 2.58\sigma_v$을 택하는 것이다. 이것은 이 범위가 비 정상적으로 큰 값과 작은 값을 배제할 수 있기 때문이다. 본 논문에서는 고정된 개수의 세그먼트를 임의로 선택하여 그들의 VPP와 EPP 값들을 계산한다. 이 값들의 분포가 가우시안 분포를 따른다고 가정함으로써 VPP에 대한 평균값 μ_{VPP}와 표준편차 σ_{VPP}, 그리고 EPP에 대한 평균값 μ_{EPP}와 표준편차 σ_{EPP}를 계산할 수 있다. 이 값들을 사용하여 $max(dom(VPP))$와 $max(dom(EPP))$를 구하면 각각 다음과 같다.

$$max(dom(VPP)) = \mu_{VPP} + 2.58 \cdot \sigma_{VPP},$$
$$min(dom(VPP)) = \mu_{VPP} - 2.58 \cdot \sigma_{VPP}$$
$$max(dom(EPP)) = \mu_{EPP} + 2.58 \cdot \sigma_{EPP},$$
$$min(dom(EPP)) = \mu_{EPP} - 2.58 \cdot \sigma_{EPP}$$

따라서, 두 세그먼트 SEG_q와 SEG_t간의 VPP와 EPP에 대한 유사도 sim_{VPP}와 sim_{EPP}는 각각 다음 식이 된다.

$$sim_{VPP}(SEG_q, SEG_t) = 1 - \frac{|VPP_q - VPP_t|}{5.16\sigma_{VPP}}$$
$$sim_{EPP}(SEG_q, SEG_t) = 1 - \frac{|EPP_q - EPP_t|}{5.16\sigma_{EPP}}$$

위의 두 가지 유사도를 결합하여 두 세그먼트 SEG_q와 SEG_t간의 기하학적 유사도 sim_G를 다음 식으로 표현할 수 있다.

$$sim_G(SEG_q, SEG_t) = \frac{w_{VPP} \cdot sim_{VPP} + w_{EPP} \cdot sim_{EPP}}{w_{VPP} + w_{EPP}}$$

여기에서, w_{VPP}와 w_{EPP}는 각각 유사도 sim_{VPP}와 sim_{EPP}에 대한 가중치이다. 이 가중치는 어플리케이션 영역에 따라 사용자에 의해 선택된다.

4.2.3 통합 유사성 척도(Integrated similarity measure)

유사성 검색을 위하여 단 하나의 유사성 척도 만을 사용하는 것은 충분한 정보를 제공하지 못한다. 따라서, 본 논문에서는 검색의 정확성을 위하여 앞 절에서 언급한 두 가지 부류의 유사성 척도를 결합하여 통합된 유사성 척도를 사용한다. 두 세그먼트 SEG_q와 SEG_t간의 통합된 유사성 척도 sim_I는 방향과 기하학적 유사성 척도를 고려하여 다음과 같이 정의된다.

$$sim_I(SEG_q, SEG_t) = \frac{w_D \cdot sim_D + w_G \cdot sim_G}{w_D + w_G}$$

여기에서, w_D와 w_G는 각각 유사도 sim_D와 sim_G에 대한 가중치이다. 이 가중치에 대한 생략시 값(default value)은 1이며, 사용자는 어플리케이션 영역에 따라 가중치를 선택할 수 있다. 유사성 검색은 일반적으로 주관적 (subjective)인 측면이 강하고, 이 가중치 값을 다양하게 부여함으로써 사용자의 요구 사항을 반영할 수 있다. 본 논문에서는 모든 유사성 척도에 대하여 생략시 값인 1을 사용한다.

4.3 유사성 검색 매커니즘

이 절에서는 거리 함수와 의미적 유사성 척도를 사용한 유사성 검색 매커니즘에 대하여 논의하기로 한다. 질의는 보통 다음의 형태로 주어진다.

Given : 질의 세그먼트 및 유사성 임계값(similarity threshold) $\zeta (0 \le \zeta \le 1)$

Target : 데이터베이스로부터 임계값 ζ범위 안에 있는 유사한 세그먼트들을 검색

일반적인 유사성에 기반한 패턴 검색에서, 질의는 시퀀스

형태로 주어지고 시퀀스 데이터베이스로부터 유사한 시퀀스나 서브 시퀀스를 검색하는 형태로 질의가 처리된다. 그러나, 본 논문에서는 검색 매커니즘을 질의 세그먼트에 대하여 유사한 세그먼트를 찾는 것으로 제한한다(실제로, 세그먼트는 서브 시퀀스로 간주될 수 있다). 이것은 본 논문에서는 제안한 유사성 척도를 검증하는 데 중점을 두기 때문이다. 본 논문에서 제안한 유사성 척도를 사용함으로써 검색 매커니즘은 시퀀스 형태로 주어지는 유사성 검색에도 어렵지 않게 확장될 수 있다.

질의가 처리되기 전에 다차원 데이터 시퀀스를 세그먼트로 분할하고, 유사성 비교를 위하여 각 세그먼트로부터 특성 값을 추출하고, 이들을 데이터베이스에 저장하는 전처리 단계(pre-processing step)가 필요하다. 알고리즘의 입력 파라미터로는 질의 세그먼트와 유사성 임계값 ζ이 있다. 질의 세그먼트로부터 역시 마찬가지로 특성 값을 추출하고, 유사성 임계값을 그에 상응하는 적절한 거리 임계값(ε)으로 변환한다. 검색 데이터 공간은 n-차원의 큐브 $[0,1]^n$로 정규화되어 있기 때문에 ζ-값을 ε-값으로 변환하는 일은 매우 단순하다.

Algorithm ***Similarity-retrieval***
Input : a query segment SEG_Q and a similarity threshold ζ
Output : a set ANS_{SEG} of answer segments
Step 0 : /* Initialization */
$ANS_{SEG} \leftarrow \varphi$ /* an answer set */
Transform a similarity threshold ζ to a distance threshold ε
Extract features from a query segment
Step 1 : /* Filtering by the distance measure D_{seg} */
for each segment SEG_i in a database
if $D_{seg}(SEG_Q, SEG_i)$ **then**
$ANS_{SEG} \leftarrow ANS_{SEG} \cup \{SEG_i\}$
Step 2 : /* Refinement by sim_I */
for each SEG_j in the set ANS_{SEG}
if $sim_I(SEG_Q, SEG_j) < \zeta$ **then**
$ANS_{SEG} \leftarrow ANS_{SEG} - \{SEG_j\}$
Step 3 : **return** set ANS_{SEG}

(그림 5) 알고리즘 *Similarity-retrieval*

Step 1은 거리 함수 D_{seg}를 사용한 필터링 단계이다. 이 단계에서는 정리 6에서 보인 바와 같이 'no false dismissal'을 보장하면서 질의와 유사하지 않은 시퀀스들을 데이터베이스로부터 배제시킨다. 비록 D_{seg}가 정확성(correctness)을 보장하긴 하나 수많은 'false hit'로 인해 검색 효율이 저하된다. 이 과오 선택은 차후의 정제(refinement) 단계에서 다시 검토해야 하는데 이는 상당한 프로세스 오버헤드를 야기시킨다. 물론 통합 유사성 척도인 sim_I를 사용하는 것은 정확성을 보장하지는 않는다. 그러나, 다음 절의 실험 결과가 나타내듯이 sim_I의 사용은 상당한 Recall율을 유지하면서 좋은 Precision율을 보여준다.

5. 실험 및 평가

5.1 실험 환경

제안한 유사성 척도의 정확성을 측정하기 위해서 실제 비디오 뿐만 아니라 가상 데이터를 사용하여 복합적인 실험을 수행하였다. 가상데이터를 사용하는 이유는 실제데이터가 다양한 크기와 매개변수를 갖도록 실험 환경을 준비하는 데에 제약이 있는 반면 가상적으로 생성된 데이터는 이러한 측면에서 더 많은 유연성을 제공하기 때문이다. 본 논문에서 제시한 유사성 척도들은 다차원 데이터 시퀀스로 표현하기가 용이한 비디오 데이터 대해서 평가되었다. 하지만 본 논문에서 제시한 유사성 척도들은 비디오 데이터에만 한정되는 것은 아니다. 시계열 데이터나 신호 처리(signal processing) 데이터 등과 같이 '시퀀스'로 표현될 수 있는 데이터들에 대해서도 본 논문에서 제시한 유사성 척도들을 사용할 수 있다. 가상 데이터에 대해서도 제시한 척도들을 평가한 이유 중의 하나도 이런 가능성을 보여주기 위한 것이다. 주요 평가 파라미터들은 precision과 recall이다.

실험상의 편의를 위해, 3차원 데이터들을 사용했으나, 제안한 방법은 데이터의 차원을 제한하지 않으며 임의의 차원의 데이터 시퀀스들도 사용될 수 있다. 합성된 데이터 시퀀스들은 다음과 같이 프렉탈 함수를 사용하여 생성하였다. 비디오 데이터는 TV 뉴스와 드라마, 그리고 다큐멘터리들로부터 추출한 시퀀스의 집합이다. 각 MDS는 비디오 클립으로부터 생성된 것이고 이 비디오 클립의 각 프레임은 3차원 공간 상에 점으로써 표현되어진다. <표 1>은 사용된 테스트 데이터를 요약한 것이다.

〈표 1〉 실험에 사용된 테스트 데이터

	가상 데이터	비디오 데이터
세그먼트 수	11,300	8,341
질의 수	50	50
유사성 임계값 ()	0.5~0.9	0.5~0.9

5.2 실험 결과 및 분석

실험의 평가는 일반적으로 정보 검색 분야에서 널리 알려진 precision과 recall에 관하여 평가하였다. 질의에 의해 검색된 세그먼트들의 집합을 Set(ret), 그리고 질의에 대한 세그먼트들의 실제 해집합을 Set(rel)로 하자. 집합 S의 요소들의 수를 |S|라 할 때, precision과 recall은 다음과 같이 정의된다.

$$precision = \frac{|Set(ret) \cap Set(rel)|}{|Set(ret)|},$$

$$recall = \frac{|Set(ret) \cap Set(rel)|}{|Set(rel)|}$$

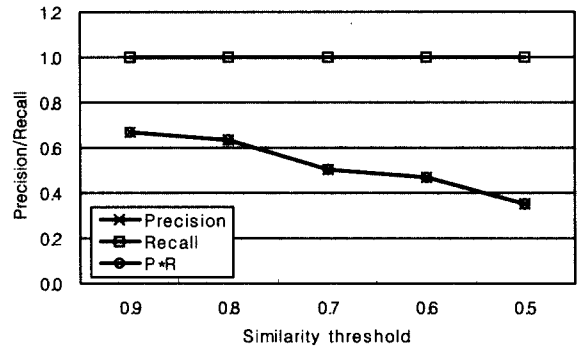
precision과 recall 둘 다를 함께 고려한 효과를 관찰하기

위해, 우리는 이 둘을 통합한 척도에 의한 평가도 포함시켰다. 이것은 precision과 recall의 곱(P*R)으로 나타낸다. 본 논문에서 제시한 유사성 척도들의 효과를 측정하기 위해서 검색의 결과를 평가하는데 기초가 되는 정답 해(ground truth)를 정의할 필요가 있다. 비디오의 경우, 많은 프레임들로 이루어진 방대한 정보를 포함하고 있다. 그렇기 때문에 사람이 각 비디오 세그먼트들을 보면서 질의 세그먼트와 얼마나 유사한지 결정하기란 쉽지 않다. 따라서, 본 논문의 실험에서는 유사성 척도 D_S를 사용하여 순차 검색(sequential scanning)에 의해 검색된 세그먼트들의 집합을 정답 해로 간주한다.

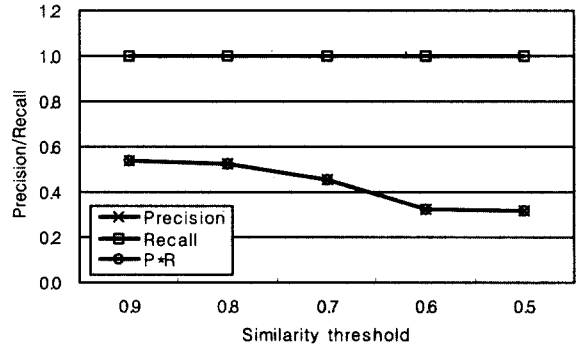
(그림 6)과 (그림 7)은 비디오 데이터와 가상으로 합성된 데이터에 대하여 다양한 유사성 임계 값에 관한 precision과 recall 그리고 P*R을 나타낸다. 이 경우에는 거리에 근거한 유사성 척도(D_{seg}) 만을 사용했다. 반면, (그림 8)과 (그림 9)는 거리에 근거한 유사성 척도 (D_{seg})와 의미에 기반한 유사성 척도(sim_I)를 함께 사용했을 때의 precision과 recall, 그리고 $P*R$를 보여준다. (그림 6)과 (그림 7)을 살펴보면 비디오 데이터와 가상 시퀀스들에 대한 recall은 모두 1이다. 이는 척도 D_{seg}가 *'no false dismissal'*을 보장하고 있기 때문이다. 그러나 precision은 비디오 데이터에 대해서는 0.35~0.37, 그리고 가상 시퀀스에 대해서는 0.31~0.54로써 상대적으로 낮다. 또한, precision은 임계 값이 증가함에 따라 감소한다. $P*R$ 값은 recall이 항상 1이기 때문에 물론 precision값과 같다.

반면에 (그림 8)과 (그림 9)에서 보여지듯이 거리에 근거한 유사성 척도(D_{seg})와 의미에 기반한 유사성 척도(sim_I)를 함께 사용했을 경우, recall은 척도 sim_I이 *'false dismissal'*을 허용하는 이유로 1이 아니다. recall은 비디오 데이터에 대해서는 0.47~0.96이고 가상 시퀀스에 대해서는 0.42~0.96이다. 그러나, precision은 비디오 데이터에 대해서는 0.77~0.85이고 가상 시퀀스에 대해서는 0.69~0.82로써, 거리에 근거한 척도만을 사용한 경우보다 훨씬 좋아짐을 알 수 있다. recall을 조금 희생함으로써 sim_I과 D_{seg}를 사용하여 너 나은 precision을 얻을 수 있다. 그리고 이것이 검색 효율성을 향상시킬 것이라는 것은 명백하다. $P*R$에 대한 관찰 결과 또한 비디오 데이터에 대해서 0.40~0.74, 그리고 가상 시퀀스에 대해서는 0.34~0.67로 훨씬 향상된 값을 보여준다.

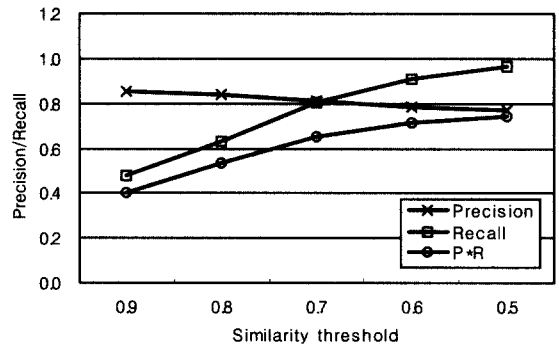
결론적으로, 거리에 근거한 유사성 척도(D_{seg})와 의미에 기반한 유사성 척도(sim_I)를 함께 사용했을 경우가 거리에 근거한 유사성 척도(D_{seg})만을 사용하는 경우 보다 비록 recall은 다소 떨어지지만 전반적으로 좋은 성능을 보인다는 것을 관찰할 수 있다. 이것은 효율(efficiency)과 정확성(correctness) 사이의 교환(trade-off)이다. 신용 카드 사용 등의 이상 감지(fraud detection)나 CCTV에 녹화된 범죄 행위의 감지와 같이 recall에 민감한 분야에서는 의미에 기반한 유사성 척도(sim_I)의 사용을 배제할 수 있다.



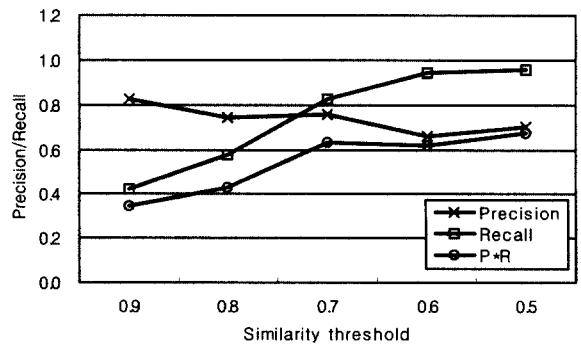
(그림 6) D_{seg}에 대한 비디오 데이터의 Precision/Recall



(그림 7) D_{seg}에 대한 가상 시퀀스의 Precision/Recall



(그림 8) D_{seg}와 sim_I에 대한 비디오 데이터의 Precision/Recall



(그림 9) D_{seg}와 sim_I에 대한 가상 시퀀스의 Precision/Recall

6. 결 론

시퀀스 데이터 세트에서의 유사성 검색은 시계열 데이터, 디지털 아날로그 신호와 같이 다양한 어플리케이션 영역에 널리 적용될 수 있기 때문에 데이터베이스 분야에서 큰 잠재력이 있는 영역 중 하나이다. 본 논문에서는 두 가지의 효과적인 유사성 척도를 제시했다. 그것은 다차원 공간에서 두 세그먼트 사이의 거리에 기초한 유사성 척도와 세그먼트의 방향성과 기하학적 특성을 고려한 의미에 기반한 유사성 척도이다. 이와 더불어, 제시된 유사성 척도들을 이용해서 데이터베이스로부터 유사한 세그먼트들을 검색하는 알고리즘을 제안하였다.

정보 검색 분야에서 폭 넓게 사용되는 precision과 recall에 대하여 제안한 유사성 척도를 평가하기위해 가상적으로 합성한 데이터와 실제 비디오로부터 추출된 데이터를 사용해 실험을 했다. 실험 결과, 상당히 의미 있는 precision과 recall값을 관찰할 수 있었다. recall은 비디오 데이터에 대해서 0.47~0.96, 그리고 가상 시퀀스에 대해서는 0.42~0.96 범위로 나타났고, precision은 비디오 데이터에 대해서 0.77~0.85이고 가상 시퀀스에 대해서는 0.69~0.82로 나타났으며, 이것은 실제 비즈니스 환경에서 유용하게 사용될 수 있다.

본 논문에서 제시된 유사성 척도는 데이터 형식이 다차원 시퀀스로 표현되어지는 다양한 어플리케이션 영역에서 폭 넓게 사용될 수 있다. 논문에서 강조된 잠재적인 어플리케이션 중 하나는 비디오 데이터 세트에서의 유사성 질의이다. 하지만 다른 어플리케이션 영역에서도 또한 사용될 수 있을 것이라 여겨진다. 향후 연구 계획으로써, 제시된 유사성 척도들을 기상의 패턴 매칭이나 주가의 변화, 그리고 음성 신호 매칭과 같은 다양한 어플리케이션 영역에 그 영역의 고유한 특성을 고려하여 적용해 보는 것에 대한 연구를 계획하고 있다.

참 고 문 헌

[1] R. Agrawal, C. Faloutsos and A. Swami, "Efficient Similarity Search in Sequence Databases, Proceedings of Foundations of Data Organizations and Algorithms (FODO)," Evanstone, Illinois, pp.69-84, October, 1993.

[2] S. Berchtold, D. Keim and H. Kriegel, "The X-tree : An Index Structure for High-Dimensional Data," Proceedings of Int'l Conference on Very Large Data Bases, India, pp. 28-39, 1996.

[3] N. Beckmann, H. Kriegel, R. Schneider and B. Seeger, "The R*-tree : An Efficient and Robust Access Method for Points and Rectangles," Proceedings of ACM SIGMOD Int'l Conference on Management of Data, New Jersey, pp.322-331, 1990.

[4] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," Proceedings of ACM SIGMOD Int'l Conference on Management of Data, Minneapolis, Minnesota, pp.419-429, 1994.

[5] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, "Query by Image and Video Content : The QBIC System," IEEE Computer, Vol.28, No.9, pp.23-32, 1995.

[6] A. Guttman, "R-trees : A Dynamic Index Structure for Spatial Searching," Proceedings of ACM SIGMOD Int'l Conference on Management of Data, Boston, Massachusetts, pp.47-57, 1984.

[7] A. Hampapur, R. Jain and T. Weymouth, "Digital Video Segmentation," ACM Multimedia, pp.357-364, 1994.

[8] A. Hinneburg and D. A. Keim, "An Efficient Approach to Clustering in Large Multimedia Databases in Noise," Int'l Conference on Knowledge Discovery in Databases and Data Mining, New York, NY, pp.58-65, 1998.

[9] D. L. Harnett and A. K. Soni, "Statistical Methods for Business and Economics," 4th Edition, Addison Wesley Publishing, 1991.

[10] E. J. Keogh, K. Chakrabarti, S. Mehrotra and M. J. Pazzani, "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases," Proceedings of ACM SIGMOD Int'l Conference on Management of Data, pp.151-162, 2001.

[11] S. L. Lee and C. W. Chung, "Hyper-Rectangle Based Segmentation and Clustering of Large Video Data Sets," Information Science, Vol.141, No.1-2, pp.139-168, 2002.

[12] S. L. Lee, S. J. Chun, D. H. Kim, J. H. Lee and C. W. Chung, "Similarity Search for Multidimensional Data Sequences," Proceedings of IEEE Int'l Conference on Data Engineering, San Diego, California, pp.599-608, 2000.

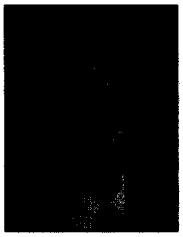
[13] D. Rafiei, "On Similarity Queries for Time Series Data," Proceedings of Int'l Conference on Data Engineering, Sydney, Australia, pp.410-417, 1999.

[14] D. Rafiei and A. Mendelzon, "Similarity-Based Queries for Time Series Data," Proceedings of ACM SIGMOD Int'l Conference on Management of Data, Tucson, Arizona, pp. 13-25, 1997.

[15] T. Sellis, N. Roussopoulos and C. Faloutsos, "The R+ Tree : A Dynamic Index for Multi-Dimensional Objects," Proceedings of Int'l Conference on Very Large Data Bases, England, pp.507-518, 1987.

[16] B. K. Yi and C. Faloutsos, "Fast Time Sequence Indexing for Arbitrary Lp Norms," Proceedings of Int'l Conference on Very Large Data Bases, pp.385-394, 2000.

[17] H. J. Zhang, J. Wu, D. Zhong and S. W. Smoliar, "An Integrated System for Content-Based Video Retrieval and Browsing, Pattern Recognition," Vol.30, pp.643-653, 1997.



이 석 룡

e-mail : sllee@hufs.ac.kr

1984년 연세대학교 기계공학과 학사

1993년 연세대학교 산업공학과 전자계산 전공 석사

2001년 한국과학기술원(KAIST) 정보 및 통신공학과 박사

1984년~1995년 한국 IBM 소프트웨어 연구소 선임연구원

2002년~현재 한국외국어대학교 산업정보시스템공학부 부교수

관심분야 : 데이터베이스, 데이터웨어하우스, 데이터마이닝, 시계열 데이터 검색

이 주 홍

e-mail : juhong@inha.ac.kr

1983년 서울대학교 컴퓨터공학과(학사).

1985년 서울대학교 컴퓨터공학과(석사).

2001년 한국과학기술원 정보및통신공학과 (박사)

1985년~1989년 한국통신 사업지원단 전임연구원

1989년~1993년 한국아이비엠 소프트웨어 연구소 선임프로그래머

2001년~현재 인하대학교 컴퓨터공학부 조교수

관심분야 : 소프트컴퓨팅과 데이터마이닝, 데이터 웨어하우스와 OLAP, 데이터베이스, 웹 서비스, 정보검색

전 석 주

e-mail : chunsj@ansan.ac.kr

1987년 경북대학교 전자공학과 컴퓨터공학 전공(학사)

1989년 경북대학교 대학원 전자공학과 컴퓨터공학전공(석사)

2002년 한국과학기술원 정보및통신공학과 (박사)

1989년~1995년 현대중공업 중앙연구소 주임연구원

1997년~현재 안산1대학 인터넷정보과 조교수

관심분야 : 데이터 마이닝, 데이터 웨어하우스와 OLAP, 멀티미디어 데이터베이스 등