Journal of Korea Multimedia Society Vol. 10, No. 8, August 2007(pp. 1017-1028)

경향 벡터 기반 비디오 스트림 검색 시스템

이석룡[†], 전석주[††]

요 약

본 논문에서는 비디오 스트림을 효과적으로 표현하고 저장하며, 저장된 비디오 스트림을 효율적으로 검색하는 기법을 제안한다. 각 비디오 프레임으로부터 특징(feature)들을 추출하고, 각 특징들의 수치값을 정규화하여 [0,1] 사이의 값으로 표현하면, f 개의 특징으로 표현된 비디오 프레임은 f 차원의 공간 $[0,1]^f$ 상의 한 점으로 나타낼 수 있다. 따라서 비디오 스트림은 다차원 공간에서 점들의 궤적으로 표현될 수 있으며 이 궤적은 카메라 샷을 기준으로 비디오 세그먼트로 분할된다. 비디오 세그먼트는 세그먼트 내의 점들의 움직임 등의 정보를 나타내는 경향 벡터(trend vector)로 표현되며, 비디오 스트림 검색은 이러한 경향 벡터에 대하여 수행된다. 스포츠, 뉴스, 기록영화, 교육용 비디오 등의 비디오 스트림에 대하여 제안한 기법을 검증하였으며, 실험 결과 기존의 방법에 비하여 복원 오차율(reconstruction error rate)이 평균37% 감소되었고, 검색의 정밀도(precision)는 비슷한 수준의 재현율(recall) 및 응답 시간을 유지하면서 평균 2.1 배까지 향상되었음을 관찰할 수 있었다.

A Video Stream Retrieval System based on Trend Vectors

Seok-Lyong Lee[†], Seok-Ju Chun[††]

ABSTRACT

In this paper we propose an effective method to represent, store, and retrieve video streams efficiently from a video database. We extract features from each video frame, normalize the feature values, and represent them as values in the range [0,1]. In this way a video frame with f features can be represented by a point in the f-dimensional space $[0,1]^f$, and thus the video stream is represented by a trail of points in the multidimensional space. The video stream is partitioned into video segments based on camera shots, each of which is represented by a trend vector which encapsulates the moving trend of points in a segment. The video stream query is processed depending on the comparison of those trend vectors. We examine our method using a collection of video streams that are composed of sports, news, documentary, and educational videos. Experimental results show that our trend vector representation reduces a reconstruction error remarkably (average 37%) and the retrieval using a trend vector achieves the high precision (average 2.1 times) while maintaining the similar response time and recall rate as existing methods.

Key words: Video Representation(비디오표현), Video Retrieval(비디오검색), Multimedia System(멀티미디어시스템)

※ 교신저자(Corresponding Author) : 이석룡, 주소 : 경기도 용인시 처인구 모현면 왕산리 산89번지(449-791), 전화 : 031)330-4357, FAX : 031)330-4357, E-mail : sllee@hufs.ac.kr
접수일 : 2007년 5월 8일, 완료일 : 2007년 7월 5일
[†] 정회원, 한국외국어대학교 산업정보시스템공학부
[††] 서울교육대학교 컴퓨터교육과 (E-mail : chunsj@snue.ac.kr)
※ 이 논문은 2004년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2004-041-D00665).

1. 서 론

비디오 스트림은 교육, VOD(video on demand), NOD(news on demand), 디지털 도서관 및 인터넷 방송 (Internet broadcasting) 등의 다양한 응용 분야에서 점점 중요한 컨텐츠로 부각되고 있으며, 비디오 서버는 비디오 스트림들을 특정 형태로 저장하고 있다가 클라이언트의 요청이 있을 때에 실시간 제약사항(realtime constraint)을 만족시키면서 사용자가 원하는 비디오 스트림들을 검색하여 전송해 주는 시스템이다. 이러한 관점에서 비디오 스트림 서버는 다수의 사용자에게 동시에 서비스를 제공해야 하므로 사용자의 요구에 대하여 일정 기준에 맞는 신속한 서비스를 제공해야 한다. 대용량, 그리고 비정형성은 비디오 스트림 데이터의 주요한 특성이며, 이러한 특성은 대용량의 저장 공간과 고속의 통신 및 컴퓨터 처리 능력을 필요로 하기 때문에 그 응용 분야가 매우 다양함에도 불구하고 다루기 어려운 분야 중의 하나이다. 이러한 어려움을 극복하기 위하여 비디오 스트림을 효과적으로 표현하고, 저장하고, 또한 검색하는 기술은 필수적이다. 본 논문에서는 비디오 스트림 서버에서 대용량 스트림 데이터를 효과적으로 저장하고 검색하기 위하여 비디오 스트림을 효과적으로 표현하고, 이에 대한 의미 기반 검색(semantic-based retrieval)을 가능하게 하는 비디오 스트림 검색 시스템을 제시한다.

비디오 스트림은 연속된 프레임들로 구성되며 각각의 프레임은 색상, 질감, 모양과 같은 다양한 특징들로 특성화된다. 하나의 프레임은 특징 공간(feature space)에서 다차원 벡터로 표현될 수 있으며, 따라서 비디오 스트림은 각 특징이 하나의 차원을 만들어 다차원 데이터 공간에서 점들의 궤적으로 모형화될 수 있다. 즉, 프레임으로부터 추출한 특징이 f 개라 할 때 N 개의 프레임으로 이루어진 비디오 스트림은 다음 식1과 같이 f-차원 공간에서 N 개의 점으로 이루어진 궤적 S_N^f 로 표현할 수 있다[1]:

$$S_N^f = \langle\langle S^f[0], S^f[1], \cdots, S^f[N-1] \rangle\rangle \quad (1)$$

여기에서 각 벡터 $S^f[i]$ $(0 \le i \le N-1)$ 는 f 개의 요소로 이루어진 벡터량으로써 f 개의 특징 값으로 표현된 비디오 프레임을 나타낸다. 이러한 비디오 스트림은 카메라 샷에 따라 수초에서 수십 초 단위의 비디오 세그먼트로 분할된다. 다음 그림 1은 비디오 스트림의 구조를 나타내고 있다.

본 논문에서는 비디오 프레임을 다차원 벡터로 표현함으로써 비디오 스트림을 다차원 공간에서 점들의 궤적으로 나타내고, 이 궤적을 카메라 샷을 기준으로 하여 비디오 세그먼트로 분할 한 후 각각의 세그먼트를 연속된 프레임들의 동적 특성을 고려하여 경향 벡터로 표현하였다. 기존의 대부분의 내용 기반 검색(content-based retrieval) 방식이 주로 비디오 프레임의 색상과 질감, 모양 등의 정적인 특징(static feature)을 추출하여 검색에 이용한 데 비하여, 정적인 특징 뿐 아니라 연속된 프레임의 움직임의 방향성(direction property)과 같은 의미 정보를 검색에 이용함으로써 검색의 정확성을 높일 수 있도록 하였다. 비디오 스트림 검색을 위하여 각 비디오 세그먼트의 경향 벡터간의 유사성 함수(similarity function: sim)를 제안하고 이를 검색에 이용하였다. 본 논문에서 다루는 비디오 스트림 검색 문제는 비디오 세그먼트의 집합 VS_SET 으로부터, 질의 비디오 세그먼트 VS_Q 와 유사성 임계 값 ξ $(0 \le \xi \le 1)$ 이 주어진 가운데, VS_Q 와 유사한 비디오 세그먼트들(VS_i)의 집합 (VS_{ANS}_SET)을 검색하는 것 (Find VS_i such that $\{VS_i \mid VS_i \in VS_{ANS}\_SET, sim(VS_Q, VS_i) \le \xi\}$) 이다. 경향 벡터를 사용한 비디오 스트림 검색 시스템의 전체적인 구성이 그림 2에 도시되어 있다. 먼저, 비디오 스트림으로부터 특징 추출 과정을 거쳐 다차원 궤적을 형성하고, 이 궤적을 비디오 샷을 기준으로 비디오 세그먼트로 분할한다. 지면상의 제약 때문에 본 논문에서는 세그멘테이션 알고리즘을 상세하게 기술하지는 않는다. 특징 추출 과정 및 세그멘테이션의 조건, 알고리즘과 같은 자세한 사항들에 대해서는 [2]를 참고하기 바란다. 다음으로, 세그멘테이션 과

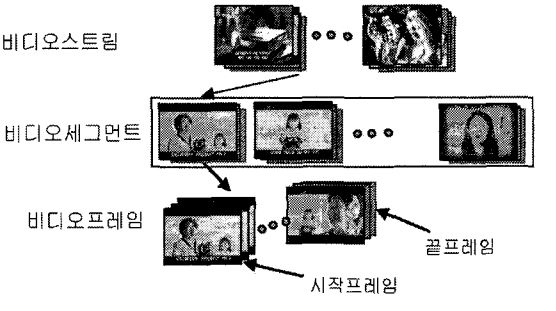


그림 1. 비디오 스트림의 구조

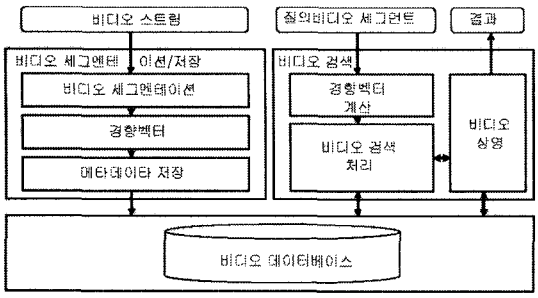


그림 2. 비디오 스트림 검색 시스템의 전체적인 구성

정에서 생성된 각 비디오 세그먼트로부터 그 세그먼트의 다양한 특성들을 고려하여 경향 벡터가 계산된다. 이것은 한 세그먼트내의 점들이 가지는 값과 점들의 움직임의 경향을 요약하여 표현하며 비디오 검색을 위해 사용되며, 경향 벡터를 내포(minimum bounding)하는 하이퍼 사각형을 나중의 검색 과정을 위해 데이터베이스에 인덱싱하고 저장한다. 이를 위해 R-트리[3]나 이와 유사한 공간 인덱싱 구조들[4-6]을 사용한다. 제안한 방법은 인덱싱 구조의 종류를 제한하지는 않으나, R*-트리[5]를 사용한다. 이것은 R*-트리가 다양한 응용 도메인에서 널리 사용되고 있는 검증된 인덱싱 구조이기 때문이다. 다음으로, 경향 벡터의 특성을 고려하여 비디오 세그먼트 사이의 유사성 척도가 정의된다. 비디오 세그먼트 사이의 유사성 척도를 사용하여 데이터베이스로부터 질의에 무관한 시퀀스들을 제거하여 후보 비디오 스트림 집합을 구하고, 정제 과정을 거쳐 이러한 후보 스트림들로부터 결과 스트림들을 얻으며, 이를 화면에 표시한다.

본 논문의 구성은 다음과 같다. 2장은 기존의 비디오 스트림의 검색 방법에 대한 고찰을 제공한다. 제안한 방법은 3장에서 자세히 기술된다. 이 장에서는 본 논문에서 사용된 비디오 유사성 척도의 명시적 정의 및 비디오 스트림 검색 알고리즘이 포함된다. 4장은 여러 가지 분야의 비디오 스트림에 대한 실험 결과를 기술하고, 5장에서 결론을 맺는다.

2. 관련 연구

1990년대 중반에 IBM Almaden 연구소에서 발표된 QBIC (Query by Image and Video Content)[7]은 내용 기반 영상/비디오 검색 시스템으로 내용 기반 검색의 효시가 된 시스템이며, 이후에 많은 대학, 연구소 등지에서 내용 기반 검색 시스템이 활발하게 연구되고 있다. QBIC은 대용량의 영상과 비디오 데이터를 검색하기 위해 공간 접근 방법의 하나인 R*-트리[5]를 사용하여 색상 및 질감과 같은 낮은 차원의 특성 데이터들을 색인하고 있다. '모양'과 같은 높은 차원의 특성 데이터들은 K-L변환을 통해 2차원이나 3차원으로 축소한 후 R*-트리를 사용하여 색인하고 있다. 즉, R*-트리를 사용하기 위해 고차원의 데이터를 저차원의 데이터로 변환한다. 이 시스템은 영상의 색상이나 질감, 모양 등과 같은 기본적인 특징에 기반한 다양한 시각적 질의를 제공한다. 또한, 데이터베이스에 저장된 영상에 대하여 사람이 의미 정보를 부여할 수 있도록 허용함으로써 제한적이나마 의미 정보에 의한 검색도 가능하다. 그러나 기본적인 특징과 의미 정보를 표현하는 논리적인 특징이 체계적으로 통합되어 있지 않으며, 다양하고 복잡한 질의 인터페이스가 체계적으로 통합되어 있지 않아서 오히려 사용자 측면에서는 이용하기가 어려울 수 있다는 단점이 존재한다.

이후에도 비디오 검색을 위하여 여러 가지 기법들[8-11]이 제안되었으며 이러한 방법들은 대표 프레임에 기반한 방법들이다. 이 방법들에서는 비디오 프레임들 중 대표 프레임을 선택하고 대표 프레임으로부터 색상, 질감, 모양 등의 특성을 추출하여 데이터베이스에 저장하고, 비디오 검색시 질의 비디오로부터 추출한 대표 프레임의 특성과 데이터베이스에 저장된 특성을 매칭함으로써 원하는 비디오를 검색한다. 이러한 방법에서는 검색 매커니즘이 대표 프레임에만 의존하기 때문에 질의 비디오와 유관한 비디오가 검색 결과 집합에서 제외되는 과오 누락 (false dismissal) 현상이 자주 발생하는 문제가 있다. Jain et al. [12]은 비디오 스트림으로부터 일정 간격으로 표본추출하여 대표 프레임을 복수 개 선정하여 이들의 특성을 데이터베이스에 저장하고 검색에 이용하는 방법을 제시하였다. 이 방법은 표본추출 비율을 높임으로써 과오 누락을 감소시킬 수 있지만 표본추출 비율이 높아질수록 저장해야 되는 프레임의 수가 증가하여 검색 효율이 떨어지는 단점이 있다. [13]에서는 움직임(motion)에 근거한 비디오 스트림 검색 방법을 제안하였다. 비디오를 다차원 공간상에서 궤적(trajectory)의 시퀀스로 표현한 후 이 시퀀스로부

터 특이점을 추출한 후, 이 특이점으로부터 커브 피팅 방식을 적용하여 비디오 스트림을 표현하였다. 이 방식은 계산 시간이 상당히 소요되어 실시간 검색과 같은 분야에서는 활용이 제한된다. 이밖에도 특정 도메인에 적합한 비디오 검색 기법이 광범위하게 연구되어 왔다. 몇 가지 예를 들면 스포츠 비디오 검색 [14], 뉴스 비디오 장면 검색 [15], 비디오에서의 얼굴 인식 [16,17] 등이다. 그러나 이러한 연구들은 해당 도메인에만 적합하도록 설계된 기법에 초점을 맞추기 때문에 그 응용이 특정 분야에 제한되어 있다.

한편, 비디오 스트림은 다차원 시퀀스로 표현될 수 있으므로 시계열 데이터의 검색 기법이 비디오 검색에 이용될 수 있다. 시계열 데이터의 표현 및 검색 기법으로써 최근에 Yi 등 [18]은 시간 시퀀스를 같은 길이의 세그먼트로 나누고 세그먼트내의 점들의 평균을 저장하는 근사 기법(approximating technique)을 제안하였다. 이 기법에서는 한 시퀀스가 s 개의 세그먼트로 나누어지며 그 시퀀스는 s 개의 세그먼트 평균들을 요소로 가지는 벡터로 표현된다. [18]을 확장하여 Keogh 등 [19] 은 APCA(adaptive piecewise constant approximation)로 불리는 새로운 차원 축소 기법을 소개하였다. 이 기법에서는 하나의 시간 시퀀스를 복원 오차가 최소가 되도록 하는 길이가 다른 복수 개의 세그먼트의 집합으로 근사하여 표현하였다. 또한, APCA에서는 각각의 길이가 다른 세그먼트가 다차원 인덱스 구조로 인덱싱될 수 있음을 보였고, D_{LB} 와 D_{AE}, 즉 하한 유클리드 거리 근사(lower bounding Euclidean distance approximation)와 비하한 유클리드 거리 근사(non-lower bounding tight approximation)를 제공하는 두 개의 거리 측정 기준을 제안하였다.

그러나, 기본적으로 이러한 기법들은 모두 일차원 시계열 데이터를 위한 유사성 검색 기법들이다. 따라서 이들은 비디오 검색에 응용될 수 있는 다차원 시퀀스에는 근본적으로 적용하기가 어렵고, 또한 위 기법들의 근사 방법에서는 세그먼트를 단순히 점이나 최소 경계 사각형으로 표현함으로써 세그먼트 내의 점들의 움직임에 관한 중요한 정보를 상실하게 된다는 점이다. 또 다른 문제점으로써, 평균에 기초한 접근 방법들[18, 19]은 원래의 시퀀스에 대하여 본질적으로 높은 복원 오차를 초래하게 된다. 이 방법에서 세그먼트 내의 점들의 움직임에 관한 정보를 보존하고 복원 오차를 줄이기 위해 세그먼트의 크기를 매우 작게 유지하는 대안을 고려할 수 있지만, 이렇게 하면 평균 값들의 수 (즉, 세그먼트의 수)가 늘어나게 되고 결국 심각한 처리 오버헤드를 초래하게 된다. Lee등은 [1]에서 두 개의 하한 (low bounding) 거리 측정 기준인 D_{mbr} 과 D_{norm}에 기초한 다차원 시퀀스를 위한 유사성 검색 방법을 제안했다. 이 검색 방법에서는 D_{mbr} 과 D_{norm}을 사용하여 질의에 무관한 시퀀스들을 데이터베이스에서 여과한다. 그러나 이러한 접근법 역시 다차원 시퀀스를 위한 검색은 지원하지만 비디오 검색을 위하여 각 프레임의 움직임에 관한 경향의 정보를 검색에 이용하지는 못하고 있다. 본 논문에서는 비디오 스트림을 구성하고 있는 각 프레임의 특징 값의 변화를 분석하여 시간의 흐름에 따른 경향 벡터로 표현하고 이를 기반으로 효율적인 비디오 검색을 수행하는 알고리즘을 설계한다.

3. 비디오 스트림의 표현 및 검색

3.1 경향 벡터를 이용한 근사

비디오 세그먼트 내의 데이터 점들의 경향을 나타내기 위한 근사 선(approximation line)을 정하는 문제는 중요하다. 그림 3의 두 비디오 세그먼트 VS_1와 VS_2에서 보인 바와 같이 한 세그먼트는 그 세그먼트 내의 점들의 궤적이 이루는 곡선의 기울기가 0인 점(즉, 고점과 저점)을 기준으로 하여 복수 개의 선분으로 분할할 수 있다. 이 선분들 중에서 다른 선분 보다 많은 점을 포함하는 지배 선분(dominant line)을 선택하고 다른 선분들은 무시함으로써, 그 선분을 해당 세그먼트의 점들의 경향으로 간주할 수 있다. 그러나 이러한 접근법은 지배 선분이 실제로는 다른 선분에 비하여 점의 수가 현저하게 많지 않는 경우가 흔히 발생할 수 있으므로 바람직하지 못하다. 보다 바람직한 선택으로 세그먼트의 시작 점과 끝 점을 잇는 선분을 지배 선분으로 선택할 수 있다. 이 경우에는 이 선분이 세그먼트 내의 모든 선분들을 연결하여 생성되는 벡터 합(vector sum)과 동일하게 되므로 보다 합리적인 선택이 된다. 따라서, 세그먼트의 시작 점과 끝 점 (Start-End)을 연결하여 생성되는 벡터 SE의 기울기를 그 세그먼트 내의 점들의 움직임의 경향을 나타내는 것으로 간주한다.

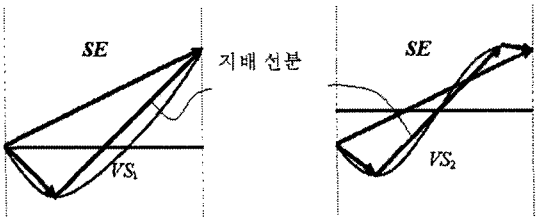


그림 3. 세그먼트 내의 점들의 움직임을 대표하는 선분의 결정

그러나 벡터 SE를 세그먼트를 대표하는 특징으로 채택하는 것은 비록 그것이 점들의 이동 경향을 잘 표현하고 있지만 적절하지 못하다. 왜냐하면 이 방식은 세그먼트의 시작점과 끝점만을 고려하고 중간점들은 전혀 고려하지 않으므로 SE에 평행하면서 복원 오차율을 최소로 하는 최적 벡터(optimal vector) OP에 비하여 일반적으로 복원 오차율이 높기 때문이다.

그림 4에서 도시한 것 과 같이 두 세그먼트 VS_3과 VS_4에서 최적 벡터 OP를 계산할 수 있다. 그러나 벡터 OP를 계산하는 것은 계산 오버헤드를 야기한다. 따라서, 세그먼트를 대표하기 위한 OP에 근사한 벡터로써 벡터 SE에 평행하면서 세그먼트의 평균선(mean line)의 중간을 지나는 벡터를 채택하고, 이를 경향 벡터(trend vector: TV)라 한다. 평균 선의 중간을 지나는 벡터를 채택하는 것은 다음과 같은 이점이 있다. 먼저, 4장의 실험에서 보인 바와 같이 이러한 방식은 복원 오차율을 획기적으로 감소시킨다. 다음으로, TV를 계산하는 것은 단순하고 매우 빠르다. 그림 4는 일 차원 시계열 데이터의 SE와 TV를 보여주고 있다. 그림에서는 편의상 시간을 수평축으로 나타냈기 때문에 평균이 선으로 표현되었지만, 일반적으로 다차원 시퀀스의 경우 시간 축을 사용하지 않으므로 평균이 다차원의 점(평균 점)으로 표현된다. 다차원 시퀀스의 경우에도 마찬가지로 벡터 TV는 벡터 SE에 평행하고 세그먼트의 평균 점을 통과한다. f-차원 공간에서 경향 벡터 TV 는 다음과 같이 정의된다.

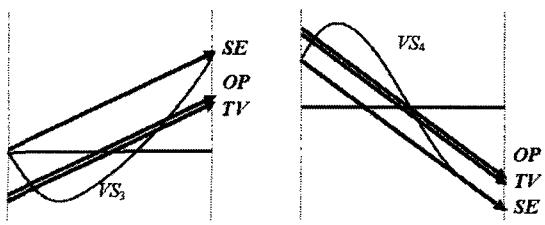


그림 4. 세그먼트의 시작-끝 벡터(SE), 최적 벡터(OP) 및 경향 벡터(TV)

$$TV = <\mathrm{k},\ m,\ a> \tag{2}$$

여기에서 k는 세그먼트 안에 포함되어 있는 점의 수이며, $m=(m_1, m_2, \cdots, m_f)$은 각 차원에서의 데이터 점들의 평균으로 구성되어 있는 세그먼트의 평균 벡터이고, $a=(a_1, a_2, \cdots, a_f)$는 세그먼트의 시작 점으로부터 끝 점을 잇는 선분의 각 차원에서의 기울기를 나타내는 기울기 벡터이다. 그림 5에서와 같이 k개의 점 $P_0, P_1, \cdots, P_{k-1}$을 포함하고 있는 비디오 세그먼트 VS에서, 세그먼트-평균 표현 방식[18,19]에 따르면 세그먼트 내의 모든 점들은 하나의 값, 즉 평균 값으로 근사하게 된다. 반면에 경향 벡터를 사용하게 되는 경우에는 세그먼트의 각 점 $P_0, P_1, \cdots, P_{k-1}$은 경향 벡터 선 상의 각 대응점인 $P_0', P_1', \cdots, P_{k-1}'$으로 근사된다. 차원 F에 대하여 점 P_i와 P_i'의 값을 각각 x_i 와 x_i' 라 하자.

f-차원의 공간에서 세그먼트-평균 표현 방식에 의한 복원 오차율(RE: reconstruction error)은 다음 식으로 나타낼 수 있다.

$$RE_{mean} = \frac{1}{k} \cdot \sum_{i=0}^{k-1} \sqrt{\sum_{j=1}^{f} \left(x_{i,j} - m_j\right)^2} \tag{3}$$

한편, 벡터 TV의 기울기 a는 식 $a = (x_{k-1}-x_0)/(k-1) = (x_{k-1}'-x_0')/(k-1)$로 주어진다. 벡터 TV에 의해 표현되는 직선의 방정식을 $f(i) = a \cdot i + \beta$ 라 하면, 이 직선이 점 $((k-1)/2, m)$ 을 지나게 되므로 $\beta = m - a(k-1)/2$ 을 얻을 수 있다. 따라서 차원 j에 관한 식은 다음과 같이 된다.

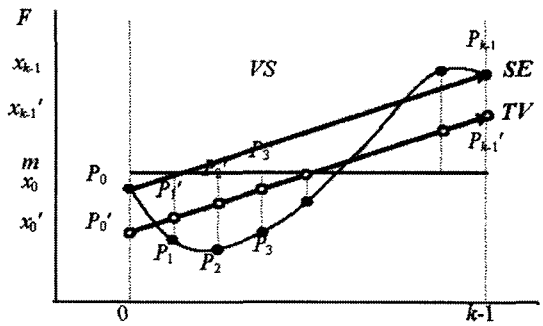


그림 5. 세그먼트-평균과 경향 벡터에 의한 세그먼트의 근사

$$x_{i,j}' = \alpha_j \cdot i + \left(m_j - \alpha_j \cdot \frac{(k-1)}{2} \right) \quad (4)$$

그러므로 d-차원의 공간에서 TV 벡터 표현 방식에 의한 복원 오차율은 다음 식으로 정의될 수 있다.

$$RE_{TV} = \frac{1}{k} \cdot \sum_{i=0}^{k-1} \sqrt{\sum_{j=1}^{f} \left(x_{i,j} - x_{i,j}' \right)^2} \quad (5)$$

3.2 메타 데이터 저장

효율적인 검색을 위하여 각 시퀀스의 세그먼트들을 공간 인덱스 구조를 사용하여 인덱싱하고 데이터베이스에 저장한다. 세그먼트의 인덱싱 방안에 관해서는 여러 가지 선택이 있을 수 있다. 그림 6은 2 차원 공간에서 이러한 선택들을 보여주고 있다. 먼저, 세그먼트 내의 모든 점들을 내포하는 최소 경계 사각형(MBR: minimum bounding rectangle)을 선택하는 것이다 [20]. 이 경우, 세그먼트는 두 점 (min_1, max_1)과 (min_2, max_2) 로 표현할 수 있다. 이 선택은 널리 알려져 있는 방식이며 '*no-false dismissal*' 의 특성을 유지한다. 그러나 이 선택은 수많은 '*false hits*'가 발생할 가능성이 있으며, 또한 <TV> 이외에도 두 점 (min_1, max_1) 과 (min_2, max_2) 을 저장해야 되는 저장 오버헤드가 있으므로 바람직한 선택은 아니다. 다음으로, 그림 6에서 보인 바와 같이 벡터 TV 자체를 최소한으로 포함하는 사각형을 선택하는 방안이다. 이 선택은 비록 '*no-false dismissal*'을 보장하지는 못하지만, 세그먼트 내의 모든 점을 포함하는 MBR에 비하여 면적이 상당 부분 줄어드는 효과가 있다. 다차원 검색 공간에서 저장되는 MBR의 부피가 작아지면 질의로 주어진 MBR과 겹칠 확률이 그만큼 줄어들게 되므로 검색 속도를 향상시킬 수 있다 [5]. 이것은 효율성과 완벽성(correctness) 사이의 타협(trade-off)이라 할 수 있으며 비디오 검색의 경우 일반적으로 완벽성을 엄격하게 요구하지는 않는다. 또한, TV 자체를 포함하는 사각형을 선택하는 방안은 이 사각형의 저점과 고점을 식 4에 의하여 간단히 계산할 수 있으므로 MBR방안과 같이 추가의 저장 공간을 필요로 하지 않는다.

3.3 비디오 세그먼트 간의 유사성

비디오 세그먼트 간의 유사성은 세그먼트 사이의

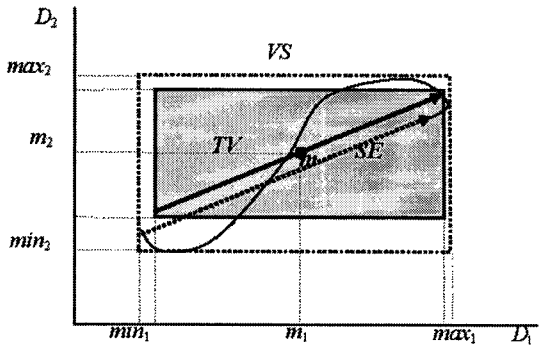


그림 6. 세그먼트를 저장하기 위한 하이퍼사각형의 선택

거리와 각 세그먼트에 대응하는 트렌드 벡터의 이동 방향의 차이라는 관점에서 정의된다. 그림 7에서와 같이 2차원 공간에서 두 개의 세그먼트, VS_1과 VS_2를 고려해 보자.

벡터 TV_1 과 TV_2의 평균점을 m_1 = ($m_{1,1}$, $m_{1,2}$), m_2 = ($m_{2,1}$, $m_{2,2}$), 두 벡터의 사이 각을 θ 라 하자. 평균에 기초한 접근 방법들에서는 두 세그먼트 간의 거리는 각 세그먼트들의 평균점 사이의 거리로써 계산된다. 따라서 f-차원의 경우에는 다음 식 6으로 표시된다.

$$d_{mean}(VS_1, VS_2) = d(m_1, m_2) = \sqrt{\sum_{j=1}^{f} (m_{1,j} - m_{2,j})^2} \quad (6)$$

경향 벡터에 의한 방법에서 두 세그먼트 간의 유사성은 거리와 방향에 기초하여 다음과 같이 정의된다.

거리에 기초한 유사성: 각각 k 개의 점을 포함하고 있는 그림 7의 비디오 세그먼트 VS_1 = ($P_{1,0}$, $P_{1,1}$,

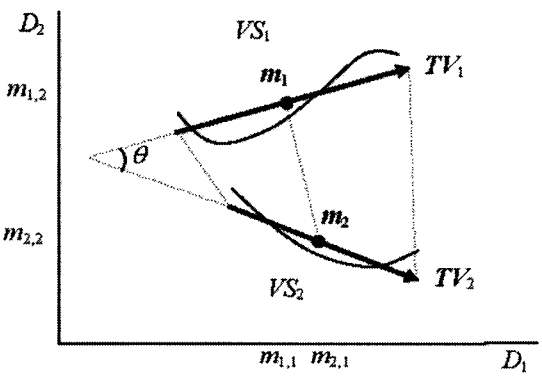


그림 7. 세그먼트 평균 (m)과 경향 벡터(TV)에 의한 두 세그먼트 간의 유사성

…, $P_{1,k-1}$) 과 VS_2 = ($P_{2,0}$, $P_{2,1}$, …, $P_{2,k-1}$) 를 고려해보자. $P_{1,i}'$ 가 점 $P_{1,i}$ 에 대응하는 경향 벡터 상의 점이고, $x_{1,i,j}'$ (식 4에 의해 계산)가 차원 j에서 점 $P_{1,i}'$의 값이라 하면, 세그먼트 VS_1 과 VS_2 는 다음의 행렬식으로 나타낼 수 있다.

$$VS_1 = \begin{pmatrix} x_{1,0,1}', & x_{1,0,2}', & \cdots, & x_{1,0,d}' \\ x_{1,1,1}', & x_{1,1,2}', & \cdots, & x_{1,1,d}' \\ & \cdots\cdots & & \\ x_{1,k-1,1}', & x_{1,k-1,2}', & \cdots, & x_{1,k-1,d}' \end{pmatrix},$$

$$VS_2 = \begin{pmatrix} x_{2,0,1}', & x_{2,0,2}', & \cdots, & x_{2,0,d}' \\ x_{2,1,1}', & x_{2,1,2}', & \cdots, & x_{2,1,d}' \\ & \cdots\cdots & & \\ x_{2,k-1,1}', & x_{2,k-1,2}', & \cdots, & x_{2,k-1,d}' \end{pmatrix}$$

두 점 $P_{1,i}'$ 와 $P_{2,i}'$간의 유클리디안 거리 $d(P_{1,i}', P_{2,i}')$ 는 다음 식 7로 주어진다.

$$d(P_{1,i}', P_{2,i}') = \sqrt{\sum_{j=1}^{f} (x_{1,i,j}' - x_{2,i,j}')^2} \tag{7}$$

그러므로 두 세그먼트, VS_1 과 VS_2 간의 거리 $d_{DS}(VS_1, VS_2)$ 는 다음의 식 8로 계산될 수 있다.

$$d_{DS}(VS_1, VS_2) = \frac{1}{k} \cdot \sum_{i=0}^{f-1} d(P_{1,i}', P_{2,i}') \tag{8}$$

길이가 서로 다른 세그먼트 간의 거리는 세그먼트 내의 점과 점이 일 대 일로 대응되지 않으므로 위의 식으로는 계산할 수 없다. 이 경우에는 짧은 길이의 세그먼트가 상대 세그먼트의 시작부터 끝까지 한 점씩 슬라이딩하면서 비교하여 거리들을 산출하고 이 중 가장 짧은 거리를 두 세그먼트 간의 거리로 채택한다. 한 편 세그먼트 VS 의 저점 P_0' 와 고점 P_{k-1}'에 의해 결정되는 하이퍼 사각형이 그림 6에서 보인 바와 같이 그 세그먼트를 인덱싱하기 위해 사용된다. 두 하이퍼 사각형 VS_1 과 VS_2 사이의 거리, $d_{HR}(VS_1, VS_2)$ 는 이 두 하이퍼 사각형 간의 최소 거리를 계산함으로써 쉽게 구할 수 있다. d_{HR} 은 데이터베이스로부터 질의와 무관한 세그먼트들을 제거하기 위하여 사용된다. 본 논문에서는 데이터 공간이 $[0,1]^f$ 하이퍼 큐브로 정규화되어 있다고 가정하므로 두 하이퍼 사각형 간의 최대 허용 거리는 하이퍼 큐브의 대각선의 길이인 $\sqrt{f}$ 가 된다. 따라서 두 세그먼트 간의 거리 $d_{DS}(VS_1, VS_2)$를 사용하여 유사도 sim_{DS} 으로 변환하면 다음 식 9와 같이 되며, 이 값은 0 과 1 사이의 값을 갖게 된다. 일반적으로 거리함수는 값이 클수록 두 세그먼트가 서로 상이함을 나타내지만 유사도는 서로 비슷할수록 1에 가까운 값을 갖고, 반대의 경우엔 0의 값에 가까운 값을 가지게 되므로, 다음의 식에서 sim_{DS} 의 값에 따라서 두 세그먼트의 유사도를 판정할 수 있다.

$$sim_{DS}(VS_1, VS_2) = 1 - \frac{1}{\sqrt{f}} d_{DS}(VS_1, VS_2) \tag{9}$$

마찬가지로 사용자가 제공한 유사도의 임계치(ξ) 역시 공간 인덱스 구조를 사용한 인덱스 검색을 하기 위하여 거리 값(δ)으로 변환하면 $\delta = \sqrt{f} \cdot (1-\xi)$ 이 된다.

방향성에 기초한 유사성: 세그먼트-평균 표현 방식의 문제점 중 하나는 검색의 효율을 감소시키는 *'false hits'* 를 많이 발생시킨다는 점이다. 이렇게 되면 수많은 세그먼트들이 차후에 세그먼트 안의 점들을 하나하나 검사하는 단계를 거쳐야 하므로 검색 비용이 증가하게 된다. 또 다른 문제점은 세그먼트-평균 표현 방식은 세그먼트 내의 점들이 어떻게 움직이는 지의 경향에 관한 정보를 전혀 가지고 있지 않다는 점이다. 두 세그먼트 VS_1 과 VS_2 간의 방향 유사성(directional similarity) sim_{DR} 은 두 벡터 TV_1 와 TV_2의 사이 각의 코사인 (일반적으로 코사인 유사성 (cosine similarity)으로 알려져 있음) 으로 정의한다. $\cos\theta$ 는 두 벡터의 내적으로 표현할 수 있으며, 코사인 유사성은 경향 벡터 TV=<k, m, a>의 a-요소를 사용하여 구할 수 있으며, 이는 a 가 TV의 기울기를 나타내기 때문이다. 예를 들어 일 차원의 경우는 a = $(x_{k-1}' - x_0')/(k-1)$ 가 된다. 따라서 f-차원의 일반적인 경우는 다음 식 10으로 나타낼 수 있다.

$$\cos\theta = \frac{\boldsymbol{TV_1} \cdot \boldsymbol{TV_2}}{|\boldsymbol{TV_1}||\boldsymbol{TV_2}|} = \frac{\boldsymbol{a_1} \cdot \boldsymbol{a_2}}{|\boldsymbol{a_1}||\boldsymbol{a_2}|} = \frac{\sum_{j=1}^{f} (x_{1,k-1,j}' - x_{1,0,j}')(x_{2,k-1,j}' - x_{2,0,j}')}{\sqrt{\sum_{j=1}^{f} (x_{1,k-1,j}' - x_{1,0,j}')^2} \cdot \sqrt{\sum_{j=1}^{d} (x_{2,k-1,j}' - x_{2,0,j}')^2}} \tag{10}$$

$\cos\theta$ 의 값의 범위는 [-1, 1] 이므로 유사성을 표현하기 위해서는 값의 범위가 [0, 1] 이 되도록 변환해야하며 이것은 다음의 식 11로 간단하게 계산될 수 있다.

$$sim_{DR}(VS_1, VS_2) = \frac{1+\cos\theta}{2} \quad (11)$$

거리 및 방향에 기초하여 두 비디오 세그먼트 사이의 유사성 $sim(VS_1, VS_2)$은 식 9와 11을 고려하여 다음과 같이 정의될 수 있다.

$$sim(VS_1, VS_2) = \frac{w_{DS} \cdot sim_{DS} + w_{DR} \cdot sim_{DR}}{w_{DS} + w_{DR}} \quad (12)$$

위 식에서 w_{DS} 는 거리를 고려한 가중치이며 w_{DR} 은 방향을 고려한 가중치이다. 사용자는 거리 요소가 상대적으로 중요하게 고려되는 경우에는 높은 w_{DS} 값을 부여하여 두 세그먼트의 유사성을 판정할 때 sim_{DS} 를 우선적으로 고려할 수 있으며, 반대의 경우에는 높은 w_{DR} 값을 부여하여 sim_{DR} 를 중요하게 고려할 수 있다. 비디오 유사성 검색은 특성상 매우 주관적이고, 따라서 가중치의 결정은 사용자의 요구 및 선호에 따라서 변경될 수 있다.

3.4. 비디오 스트림의 유사성 검색 알고리즘

이 절에서는 주어진 비디오 세그먼트와 유사한 세그먼트를 데이터베이스로부터 검색하는 알고리즘을 기술한다. 먼저 질의 세그먼트로부터 경향 벡터를 구하고 경향 벡터를 내포하는 하이퍼사각형을 계산한 후, 현 질의 세그먼트와 데이터베이스에 저장되어 있는 세그먼트에 대하여 거리의 척도인 d_{HR} 을 사용하여 후보 세그먼트를 추출하기 위하여 인덱스 (R*-tree) 를 검색한다. 발견된 후보 세그먼트들의 집합에서 질의와 무관한 세그먼트를 제거하고 원하는 비디오 세그먼트를 구하기 위하여 세그먼트 간의 유사성 척도인 sim 을 사용하여 후보 세그먼트들을 검증하여 최종 결과 세그먼트 집합을 구한다. 다음은 이 과정에 대한 알고리즘을 기술하고 있다.

Algorithm 비디오_유사성_검색
$VS_{CAND\_SET} \leftarrow \theta$ /* 후보 비디오 세그먼트의 집합 */
$VS_{ANS\_SET} \leftarrow \theta$ /* 결과 비디오 세그먼트의 집합 */
Step 0: /* 선행처리 */
각 비디오 스트림을 비디오 세그먼트로 분할
비디오 세그먼트로부터 경향 벡터 추출하고 경향 벡터와 관련된 메타데이터를 데이터베이스에 저장
Step 1: /* 질의 시퀀스의 경향 벡터 추출 */
질의 세그먼트 VS_Q 로부터 경향 벡터를 추출
Step 2: /* 1st Pruning을 위하여 d_{HR} 을 사용한 여과 과정 */
for each VS_i in a database
if $(d_{HR}(VS_Q, VS_i) \leq \delta)$
$VS_{CAND\_SET} \leftarrow VS_{CAND\_SET} \cup \{VS_i\}$
Step 3: /* 2nd Pruning을 위하여 sim 을 사용한 정제 과정 */
for each VS_i in VS_{CAND_SET}
if $(sim(VS_Q, VS_i) \geq \xi)$
$VS_{ANS\_SET} \leftarrow VS_{ANS\_SET} \cup \{VS_i\}$
Step 4: **return** VS_{ANS_SET}

4. 실험 및 고찰

4.1. 실험을 위한 준비

제안한 방법의 성능을 검증하기 위하여 TV 뉴스, 드라마 및 기록영화와 같은 비디오 스트림을 이용하여 실험을 행하였다. 실험은 원래의 비디오 스트림을 다차원 공간에 표현할 때의 궤적에 비하여 경향 벡터로 표현하였을 경우의 복원 오차율, 비디오 검색의 정밀도와 재현율, 그리고 응답시간에 대하여 이루어졌다. 실험을 위한 구현 환경으로써, 하드웨어는 Pentium IV, 2G 메모리, 160 GB HDD 가 사용되었고, 소프트웨어는 Window 2000 Server하에서 Java를 사용하였다. 실험을 위한 모든 데이터 세트는 편의상 3차원으로 하였으나 제안한 방법은 차원에 제한을 두지 않으며 다른 차원도 실험에서 사용 가능하다. 실험에 사용된 비디오 스트림은 60-1000 프레임의 다양한 길이로 구성된 1500개의 비디오 세그먼트이며 세그먼트 길이에 따라 5가지의 비디오 유형(유형1: 60-150, 유형2: 151-300, 유형3: 301-500, 유형4: 501-700, 유형5: 701-1000) 으로 분류하였고, 각 유형은 300개의 비디오 세그먼트를 포함한다. 유사성 임계 값은 0.70-0.95 의 범위에서 선택되었다. 비디오 스트림에 대한 데이터 시퀀스는 다양한 비디오 데이터 소스의 각 프레임의 픽셀로부터 색상 특징(RGB) 값을 추출하고 그들을 평균함으로 생성되었다. RGB 특징들의 값의 범위는 [0, 255]이며 이를 255로 나누면 $[0, 1]^3$ 단위 큐브 내에 포함되게 된다. 따라서, 각 프레임은 단위 큐브 내의 한 점으로 사상된다. 본 실험에서는 다음의 5가지의 경우를 비교하여 제안한 방법의 성능을 평가한다: (1) SCAN: 순차 검색

(sequential scanning)에 기초한 방법, (2) MEAN1: 실험의 편의상 같은 길이의 비디오 세그먼트로 길이를 조정한 데이터세트에서의 세그먼트-평균에 기초한 검색 방법, (3) MEAN2: 임의 길이 세그먼트에서의 세그먼트-평균에 기초한 검색 방법, (4) TV1: 실험의 편의상 같은 길이의 비디오 세그먼트로 길이를 조정한 데이터세트에서의 경향 벡터에 기초한 검색 방법, (5) TV2: 임의 길이 세그먼트에서의 경향 벡터에 기초한 검색 방법.

4.2. 실험 결과 및 고찰

복원 오차를 평가하기 위해서 제안한 방법(TV2)과 평균에 기초한 방법(MEAN2)을 서로 비교하였으며 복원 오차는 식3과 5를 이용하여 계산하였다. MEAN2와TV2에서 세그먼트를 저장하기 위한 저장 공간을 고려해보면 MEAN2에서는 세그먼트의 길이와 평균을 저장하기 위해 2개의 요소가 필요하다. 그러나, TV2에서는 이에 더하여 기울기 요소를 더 저장해야 한다. 3차원 공간의 경우를 예로 보면, MEAN2에서는 4 개의 요소를 저장해야 한다. k를 위해 1개와 m을 위해 3개의 숫자를 저장하기 위함이다. 반면 TV2에 대해서는 기울기를 고려하여 7개 요소를 저장해야 한다. 같은 저장 공간 요구 조건 하에서의 대등한 비교를 위해, TV2의 세그먼트 평균 길이를 MEAN2보다 거의 2배로 할 필요가 있다. 그림 8은 이와 같은 적절한 조정을 한 후의 MEAN2와 TV2사이의 복원 오차를 비교한 그래프이며, Y-축은 복원 오차율을 나타낸다. 이 경우에서의 실험 결과는 TV2의 복원 오차가 원래의 시퀀스에 대해서 MEAN2와 비교해서 평균 37%, 그리고 최고 48%까지 감소하였음을 관찰할 수 있다.

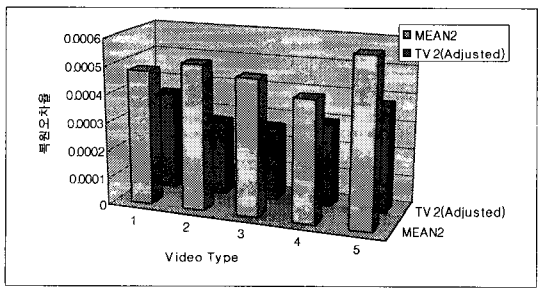
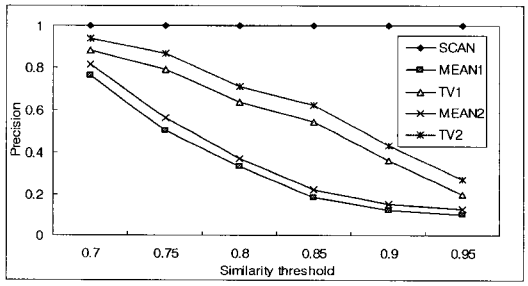


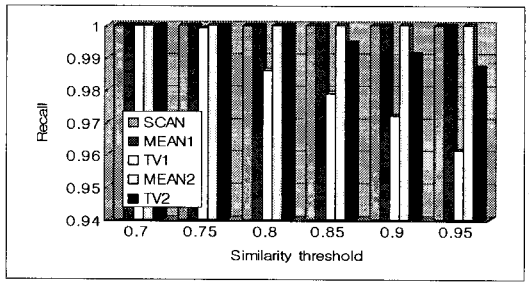
그림 8. 대등한 저장 공간 요구 조건을 고려한 MEAN2와 TV2 사이의 복원 오차 비교

다음으로 여러 분야의 유사성 검색에서 잘 알려진 정밀도와 재현율에 대하여 평가하였다. 제안한 방법의 효과를 측정하기 위해 검색의 품질을 평가하는 기준을 제공하는 그라운드 트루스 (ground truth)를 설정하는 것이 필요하다. 일반적으로 비디오 스트림은 수많은 시퀀스들로 구성되는 대용량의 정보이므로, 인간이 일일이 확인하여 두 시퀀스가 유사한 지를 결정하는 것은 쉽지 않고, 또한 이 경우에는 관찰자의 주관이 개입될 여지가 매우 높다. 본 실험에서는 그라운드 트루스를 SCAN 방법에 의해 검색된 시퀀스의 집합으로 정하였다. 즉, SCAN 방법에 의해 검색된 시퀀스들을 질의와 유사한 시퀀스로 간주한다는 의미이다.

제거(pruning) 효과는 유사성 임계 값을 변경시킴으로써 측정하였다. 유사성 값 0은 두 시퀀스가 완전히 상이함을 나타내는 반면, 1은 두 시퀀스가 서로 같다는 것을 의미한다. 실험에서 유사성 임계 값을 0.7 ~ 0.95 사이의 값으로 정하였는 데, 이는 이 범위 내에서 대부분의 제거 효과를 관찰할 수 있기 때문이다. 실험 결과, 유사성 임계 값이 감소함에 따라서 정밀도가 증가함을 보여준다. 이것은 임계 값을 작게 할수록 질의와 유사하며 검색된(relevant and retrieved) 비디오 세그먼트의 증가율이 검색된(retrieved) 세그먼트의 증가율 보다 커지기 때문이다.



(a) 정밀도



(b) 재현율

그림 9. 정밀도 및 재현율 비교

그림 9의 (a)와 (b)는 각각 5가지 방법들의 정밀도와 재현율 결과를 보여 주며, Y-축은 각각 정밀도와 재현율을 0과 1사이의 값으로 표시하고 있다. TV2의 정밀도는 0.26-0.95 분포로써 제거 효과 면에서 가장 좋은 결과를 보여 준다. 비록 제안한 방법 (TV1 과 TV2)은 정확성(correctness)을 보장하지는 못하지만 95% 이상의 재현율을 유지하며, 평균에 기초한 방법(MEAN1과 MEAN2) 보다 뛰어난 정밀도를 제공함을 관찰할 수 있다. 다음의 표 1은 평균에 기초한 방법에 대한 제안한 방법의 정밀도와 재현율의 비율을 나타낸다. 사용자는 유사성 임계 값에 따른 정밀도와 재현율의 변화를 관찰함으로써 적절한 유사성 임계 값을 선택할 수 있다. 이 실험에서는 적당한 수준의 재현율을 유지하면서 높은 정밀도를 제공하는 유사성 임계 값을 발견하는 데 초점을 맞춘다. 그림 10에서와 같이 0.85~0.90사이의 임계 값이 제거 효과 측면에서 상당한 이득을 얻을 수 있는 임계 값으로 관찰된다.

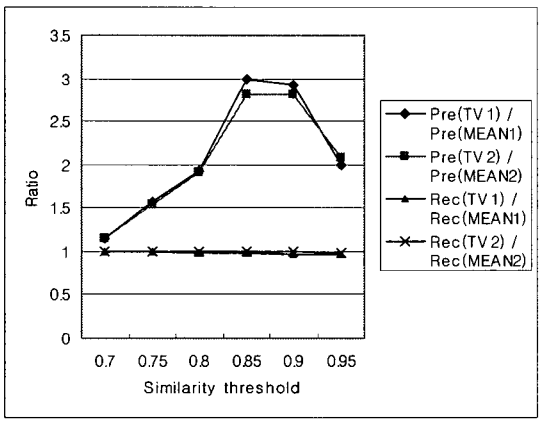


그림 10. 정밀도 이득 대비 재현율의 손실

그림 11은 SCAN 경우를 제외한4가지 경우의 평균 검색 응답 시간의 비교를 보여준다. SCAN 경우에서의 응답 시간은 너무 큰 값이므로 그림에서 생략하였다. 실험 결과에서 보듯이 TV2 대 MEAN2 의 응답 시간 비는 1.2 - 1.4 로 제안한 방법의 응답 시간이 평균에 기초한 방법에 비하여 늦은 것을 알 수 있다. 이는 검색 과정에서 평균에 기초한 방법이 한 비디오 세그먼트를 대표하기 위하여 평균 점 하나만을 사용하는 것에 비하여, 제안한 방법은 세그먼트 내의 점들의 경향을 나타내기 위하여 경향 벡터라는 선분을 사용하기 때문이다. 본질적으로 전자는 두 점을 비교하며, 후자는 두 선을 비교하기 때문에 나타나는 결과이다. 이러한 응답 시간의 증가는 제안한 방법이 달성한 정확도에 대한 보상이며, 그림 11에서 관찰되는 바와 같이 임계 값의 크기가 증가할수록 그 차이는 점점 감소함을 알 수 있다.

5. 결 론

본 논문에서는 경향 벡터 표현에 기반한 비디오

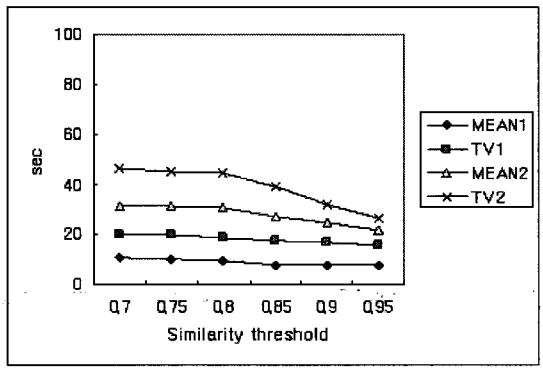


그림 11. 평균 검색 응답 시간 비교

표 1. 평균에 기초한 방법과 제안한 방법의 정밀도와 재현율의 비율

Ratio \ Threshold	0.7	0.75	0.8	0.85	0.9	0.95	Sum	Avg.
$Precision_{TV1}$ / $Precision_{MEAN1}$	1.152	1.574	1.936	2.990	2.919	2.000	12.571	2.095
$Precision_{TV2}$ / $Precision_{MEAN2}$	1.155	1.538	1.916	2.799	2.804	2.077	12.289	2.048
$Recall_{TV1}$ / $Recall_{MEAN1}$	1.000	0.999	0.986	0.979	0.972	0.961	5.897	0.983
$Recall_{TV2}$ / $Recall_{MEAN2}$	1.000	1.000	1.000	0.995	0.991	0.987	5.973	0.995

스트림의 검색 시스템에 관하여 연구하였다. 이 문제를 해결하기 위하여 비디오 스트림을 비디오 세그먼트로 분할하고, 각 세그먼트를 세그먼트 내의 점들의 움직임의 경향을 나타내는 경향 벡터로 표현하였다. 경향 벡터의 표현 방식은 보다 향상된 근사 방법을 제시한다는 점에서 기존에 제시된 평균에 기초한 방법들에 비하여 장점을 가지고 있다. 경향 벡터를 토대로 거리 및 방향성에 기초한 유사성의 척도를 정의하였으며, 이 척도를 사용하여 데이터베이스로부터 질의에 무관한 시퀀스들을 제거하여 원하는 비디오 세그먼트를 검색하는 알고리즘을 제시하였다.

제안한 방법의 성능을 검증하기 위하여 TV 뉴스, 드라마 및 기록영화와 같은 비디오 스트림을 이용하여 3 차원 환경에서 실험을 수행하였다. 실험 결과, 제안한 방법은 기존의 평균에 기초한 방법에 비하여 비슷한 수준의 재현율 및 다소 느린 응답 시간을 유지하면서, 정밀도는 평균 2.1 배, 최대 2.9 배까지 향상되었음을 관찰할 수 있었다. 또한, 원래의 시퀀스에 대한 복원 오차에 관한 실험에서 제안한 방법은 기존의 방법에 비하여 평균 37%, 최대 48% 까지 감소되었음을 보여 주고 있다.

참 고 문 헌

[1] S. L. Lee, S. J. Chun, D. H. Kim, J. H. Lee, and C. W. Chung, "Similarity search for multidimensional data sequences," *Proc. of IEEE ICDE*, pp. 599-608, 2000.

[2] 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.

[3] A. Guttman, "R-trees: a dynamic index structure for spatial searching," *Proc. of ACM SIGMOD*, pp. 47-57, 1984.

[4] S. Berchtold, D. Keim, and H. Kriegel, "The X-tree: an index structure for high-dimensional data," *Proc. of Int'l Conference on VLDB*, pp. 28-39, 1996.

[5] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: an efficient and robust access method for points and rectangles," *Proc. of ACM SIGMOD*, pp. 322-331, 1990.

[6] N. Katayama and S. Satoh, "The SR-tree: an index structure for high-dimensional nearest neighbour queries," *Proc. of ACM SIGMOD*, pp. 369-380, 1997.

[7] 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.

[8] P. Browne and A.F. Smeaton, "Video retrieval using dialogue, keyframe similarity and video objects," *IEEE Int'l Conference on Image Processing*, pp. III-1208-1211, 2006.

[9] A. Hampapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J. Bach, M. Gorkani, and R. Jain, "Virage Video Engine," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases V*, pp. 188-197, 1997.

[10] K. W. Sze, K. M. Lam, and G. Qiu, "A new key frame representation for video segment retrieval," *IEEE transactions on circuits and systems for video technology*, Vol. 15, No. 9, pp. 1148-1155, 2005.

[11] 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.

[12] A. K. Jain, A. Vailaya, and X. Wei, "Query by Video Clip," *Multimedia Systems*, Vol. 7, pp. 369-384, 1999.

[13] J.W. Hsieh, S.L. Yu, and Y.S. Chen, "Motion-based video retrieval by trajectory matching," *IEEE transactions on circuits and systems for video technology : a publication of the Circuits and Systems Society*, Vol. 16 No. 3, pp. 396-409, 2006.

[14] H.C. Shih and C.L.Huang, "Content-based scalable sports video retrieval system," *IEEE Int'l Symposium on Circuits and Systems*, Vol. 2, pp. 1553-1556, 2005.

[15] Z. Xiong, X. S. Zhou, Q. Tian, Y. Rui, and T. S. Huangm, "Semantic retrieval of video - review of research on video retrieval in meetings, movies and broadcast news, and sports," *IEEE signal processing magazine*, Vol. 23, No. 2, pp. 18-27, 2005.

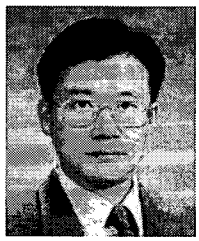
[16] D. D. Le, S. Satoh, and M. E. Houle, "Face retrieval in broadcasting news video by fusing temporal and intensity information," *Proc. of Int'l conference on image and video retrieval*, pp. 391-400, 2006.

[17] D. Xi and S.W. Lee, "Face detection and facial component extraction by wavelet decomposition and support vector machines," Audio- and Video-based Biometric Person Authentication, *Lecture Notes in Computer Science*, Vol. 2688, pp. 199-207, 2003.

[18] B. K. Yi and C. Faloutsos, "Fast time sequence indexing for arbitrary Lp norms," *Proc. of Int'l Conference on VLDB*, pp. 385-394, 2000.

[19] E. Keogh, K. Chakrabarti, S. Mehrotra, and M. J. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," *Proc. of ACM SIGMOD*, pp. 151-162, 2001.

[20] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," *Proc. of ACM SIGMOD*, pp. 419-429, 1994.



이 석 룡

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 조교수
관심분야 : 데이터베이스, 데이터웨어하우스, 데이터마이닝, 시계열 데이터 검색

전 석 주

1987년 경북대학교 전자공학과 컴퓨터공학 전공(학사)
1989년 경북대학교 대학원 전자공학과 컴퓨터공학전공(석사)
2002년 한국과학기술원 정보및 통신공학과 (박사)
현대중공업 중앙연구소 주임연구원
2003년~현재 서울교육대학교 컴퓨터교육과 조교수
관심분야 : 데이터 마이닝, 데이터 웨어하우스와 OLAP, 멀티미디어 데이터베이스