논문 2006-43CI-4-11

고차원 공간에서 효과적인 차원 축소 기법

(An Effective Method for Dimensionality Reduction in High-Dimensional Space)

정 승 도*, 김 상 욱**, 최 병 욱**

(Seungdo Jeong, Sang-Wook Kim, and Byung-Uk Choi)

요 약

멀티미디어 정보 검색에서 멀티미디어 데이터는 고차원 공간상의 벡터로 표현된다. 이러한 특징 벡터를 효율적으로 검색하기 위하여 다양한 색인 기법이 제안되어 왔다. 그러나 특징 벡터의 차원이 증가하면서 색인 기법의 효율성이 급격히 떨어지는 차원의 저주 문제가 발생한다. 차원의 저주 문제를 해결하기 위하여 색인하기 이전에 원 특징 벡터를 저차원 공간상의 벡터로 사상하는 차원 축소 기법이 제안된 바 있다. 본 연구에서는 벡터의 놈과 각도 성분을 이용하여 유클리드 거리를 근사하는 함수를 기반으로 하는 새로운 차원 축소 기법을 제안한다. 먼저, 유클리드 거리 근사를 위하여 추정된 각도의 오차의 발생 원인을 분석하고, 이 오차를 줄이기 위한 기본 방향을 제시한다. 또한, 고차원 특징 벡터를 다수의 특징 서브 벡터들의 집합으로 분리하고, 각 특징 서브 벡터로부터 놈과 각도 성분을 근사하여 차원을 축소하는 새로운 기법을 제안한다. 각도 성분을 정확하게 근사하기 위해서는 올바른 기준 벡터의 설정이 필수적이다. 본 연구에서는 최적 기준 벡터의 조건을 제시하고, Levenberg-Marquardt 알고리즘을 이용하여 기준 벡터를 선정하는 방법을 제안한다. 또한, 축소된 저차원 공간상의 벡터들을 위한 새로운 거리 함수를 정의하고, 이 거리 함수가 유클리드 거리 함수의 하한 함수가 됨을 이론적으로 증명한다. 이는 제안된 기법이 착오 기각의 발생을 허용하지 않으면서 효과적으로 차원을 줄일 수 있음을 의미하는 것이다. 끝으로, 다양한 실험에 의한 성능 평가를 통하여 제안하는 방법의 우수성을 규명한다.

Abstract

In multimedia information retrieval, multimedia data are represented as vectors in high dimensional space. To search these vectors effectively, a variety of indexing methods have been proposed. However, the performance of these indexing methods degrades dramatically with increasing dimensionality, which is known as the dimensionality curse. To resolve the dimensionality curse, dimensionality reduction methods have been proposed. They map feature vectors in high dimensional space into the ones in low dimensional space before indexing the data. This paper proposes a method for dimensionality reduction based on a function approximating the Euclidean distance, which makes use of the norm and angle components of a vector. First, we identify the causes of the errors in angle estimation for approximating the Euclidean distance, and discuss basic directions to reduce those errors. Then, we propose a novel method for dimensionality reduction that composes a set of subvectors from a feature vector and maintains only the norm and the estimated angle for every subvector. The selection of a good reference vector is important for accurate estimation of the angle component. We present criteria for being a good reference vector, and propose a method that chooses a good reference vector by using Levenberg-Marquardt algorithm. Also, we define a novel distance function, and formally prove that the distance function lower-bounds the Euclidean distance. This implies that our approach does not incur any false dismissals in reducing the dimensionality effectively. Finally, we verify the superiority of the proposed method via performance evaluation with extensive experiments.

Keywords : multimedia information retrieval, high dimensional indexing, dimensionality reduction

* 학생회원, 한양대학교 전자통신컴퓨터공학과
(Dept. of Electronics and Computer Engineering, Hnayang University)
** 정회원, 한양대학교 정보통신대학
(College of Information and Communications, Hanyang University)
※ 본 연구는 제주대학교를 통한 정보통신부 및 정보통신진흥원의 대학 IT연구센터 지원사업의 지원을 받았음.
(IITA-2005-C1090-0502-0009)
접수일자: 2006년4월18일, 수정완료일:2006년7월1일

I. 서 론

최근 멀티미디어 데이터의 급격한 증가로 인하여 효율적인 멀티미디어 정보검색에 관한 연구가 활발히 진행되어 왔다[4][10][11][19]. 멀티미디어 정보검색이란 멀티미디어 데이터베이스로부터 사용자의 질의를 만족하는 정보를 찾는 연산이다. 멀티미디어 정보검색을 위하여 많은 기존의 연구에서는 멀티미디어 데이터를 특징벡터(feature vector)의 형태로 표현한다. 특징벡터란 멀티미디어 데이터가 가지는 내용(contents) 혹은 특징들(features)을 정량화하여 벡터 형식으로 표현한 것이다. 효과적인 멀티미디어 정보검색을 위해서는 원 데이터가 가지는 정보를 최대한 반영할 수 있도록 고차원 공간상의 특징벡터를 추출해야 한다. 따라서 멀티미디어 데이터에 대한 특징벡터는 수십에서 수백 차원의 고차원 벡터의 형태로 나타난다[6][11][22][23][25].

멀티미디어 정보검색을 위해서 각 멀티미디어 데이터와 대응되는 특징벡터를 데이터베이스에 저장하게 된다. 이때, 데이터베이스에 저장된 특징벡터를 데이터 벡터(data vector)라 정의한다. 또한, 질의에 사용되는 특징벡터는 질의 벡터(query vector)라 정의한다. 멀티미디어 정보검색을 위한 사용자의 질의는 질의 벡터와의 차이가 유사 허용치(ε) 이하인 데이터 벡터들을 찾는 범위 질의(range query)와 질의 벡터와 가장 유사한 K 개의 데이터 벡터들을 찾는 K-최근접 질의(K-nearest neighbor query)로 분류된다[5][6][23]. 또한, 많은 연구들에서는 벡터간의 유사도 기준으로 유클리드 거리(Euclidean distance)를 사용하고 있다[2][6][11].

멀티미디어 정보검색의 효율을 높이기 위하여 고차원 특징벡터들을 색인할 수 있는 다양한 색인 구조(index structure)들이 제안되었다[3][4][7][14][16][25]. 그러나 고차원 특징벡터를 사용해야 하는 멀티미디어 정보검색에 적용할 경우, 특징벡터의 차원이 증가할수록 색인 구조를 이용한 검색 기법들의 성능이 급격히 떨어지는 현상이 발생한다. 이러한 현상은 차원의 저주(dimensionality curse)라고 알려져 있다[24]. 차원 축소(dimensionality reduction) 기법이란 고차원 공간상의 특징벡터를 저차원 공간상의 벡터로 변환하는 방법으로, 차원의 저주 문제를 해결하기 위한 방법 중 대표적인 방법의 하나이다[1][8][9][13][20]. 이 때 축소된 차원에서의 특징벡터를 간략히 저차원 특징벡터라 부른다.

차원 축소 기법을 이용한 멀티미디어 정보 검색은 필터링 단계 (filtering step)와 후처리 단계 (post-processing step)로 나뉘어 수행된다. 이를 이단계 검색 기법 (two-step searching method)이라 한다*. 필터링 단계에서는 축소된 저차원 특징벡터를 사용하여 사용자 유사도 조건을 만족할 가능성이 높은 데이터들만을 정답 후보(candidates)로 선택한다. 후처리 단계에서는 필터링 단계에서 선별된 정답 후보 집합을 대상으로 실제 고차원 데이터 벡터를 사용하여 정답 여부를 검증한다.

효과적인 멀티미디어 정보검색 기법은 검색 속도가 빠르면서 동시에 상세 검색(exhaustive search)과 동일한 검색 결과를 보장해야 한다. 착오 채택(false alarm)은 정답이 아니면서 필터링 단계에서 제거되지 않고 후보 집합 내에 포함됨으로써 그의 제거를 위하여 후처리 단계를 요구하는 경우를 말한다. 착오 채택의 개수가 많으면 후처리 단계에 대한 부담이 증가하여 전체 수행 시간이 증가하게 된다. 착오 기각(false dismissal)이란 정답이 필터링 단계에서 후보 집합에서 제외됨으로써 최종 정답 집합에서 제외되는 경우를 말한다. 정보검색에서 착오 기각이 발생한다는 것은 정답을 정확히 보장하지 못함을 의미하므로 매우 심각한 단점으로 지적된다.

요약하면, 바람직한 이단계 검색 기법은 필터링 단계에서 착오 채택을 최소화하면서 동시에 착오 기각이 발생하지 않음을 보장해야 한다. 착오 기각의 방지를 위해서는 저차원 특징벡터 공간에서의 두 벡터간의 거리가 원 특징벡터 공간에서 두 벡터간의 거리 보다 항상 작거나 같아야 한다는 하한 조건(lower-bound property)를 만족해야만 한다[23]. 착오 채택의 개수를 줄이기 위해서는 임의의 두 벡터들에 대하여 저차원 특징 벡터 공간에서의 거리와 원 특징벡터 공간에서의 거리의 차가 작아야 한다.

차원 축소 기법에 관한 기존의 연구들에서는 고차원 공간에서 저차원 공간으로 변환하기 위하여 다양한 수학적 변환 기법들을 사용하고 있다. 가장 일반적인 차원 축소 기법으로는 주성분 분석(Principal Component Analysis: PCA)을 이용한 기법이 있다[18]. 또 다른 기법들로는 이산 코사인 변환(Discrete Cosine Transform: DCT)을 이용한 기법과 이산 푸리에 변환(Discrete Fourier Transform: DFT)을 이용한 기법이 있다[17]. 일반적으로 PCA나 DCT를 이용한 기법이 DFT를 이용한 기법에 비하여 더 나은 성능을 보이는 것으로 알려져 있다[13].

* 이단계 검색과 달리 실제 고차원 데이터 벡터를 모두 조사하여 정답을 결정하는 방식을 상세 검색(exhaustive search)이라 한다[15]. 즉, 상세 검색은 필터링 단계를 거치지 않는다.

PCA는 데이터간의 연관성을 분석하여 연관성이 작은 고유 공간(eigen space)을 위한 고유 벡터(eigenvector)를 구하는 기법이다. 각 고유 벡터는 연관된 고유치(eigenvalue)가 존재한다. PCA를 이용한 차원 축소 기법에서는 고유치를 기준으로 선별적으로 선택된 고유 벡터를 기저 벡터(basis vector)로 사용한다. 기저 벡터를 사용하면 연관성이 높은 고차원 특징벡터를 평균 제곱 오차(mean-square error: MSE)를 최소화하면서 연관성이 적은 저차원 특징벡터로 변환할 수 있다. 여기서 기저 벡터란 저차원 공간을 형성하는 축에 해당하는 벡터를 의미한다. 특징벡터에 대한 저차원 공간에서의 각 차원 값은 원 특징벡터와 각 기저 벡터의 내적 연산을 통해 얻을 수 있다. 이 때, 기저 벡터의 개수를 선택함으로서 차원 축소 정도를 조절할 수 있다[18].

DCT를 이용한 차원 축소 기법에서는 코사인 기저 함수(cosine basis function)를 이용하여 고차원 공간의 특징벡터를 저차원 특징벡터로 변환한다. PCA를 이용한 차원 축소 기법과 같이 DCT를 이용한 차원 축소 기법에서도 기저 함수의 개수를 선택함으로써 축소할 특징벡터의 차원 수를 조절할 수 있다. 일반적으로 신호의 에너지는 저주파 성분에 집중되어 있다. 특징벡터 역시 저주파 성분에 에너지가 집중되어 있으므로 저주파 성분에 많은 정보를 담고 있다. 따라서 DCT를 이용하여 차원을 축소하고자 할 경우, 원 정보의 손실을 최소화하기 위해서는 저주파 기저 함수들을 순서대로 사용해야 한다. 저차원 공간상의 특징벡터의 각 차원 값은 DCT 기저 함수에 의해 계산되는 DCT 계수이다[13].

본 논문에서는 고차원 벡터를 위한 효과적인 차원 축소 기법에 관하여 다룬다. 착오 기각을 발생시키지 않기 위해서는 저차원 특징벡터간 거리 함수가 원 특징벡터간 거리 함수에 대한 하한이 되어야만 한다. Cauchy-Schwartz 부등식은 두 벡터의 놈(norm)의 곱이 벡터 내적보다 항상 크거나 같다는 내적에 대한 상한을 정의한다[18]. 따라서 두 벡터의 내적 항을 포함하는 유클리드 거리 함수에서, 벡터의 내적을 두 벡터의 놈의 곱으로 대체한 거리 함수는 거리식의 하한 함수로 사용될 수 있다[8][9]. 그러나 Cauchy-Schwartz 부등식을 적용한 유클리드 거리의 하한 함수는 벡터간 각도(angle) 성분을 무시하고 놈만으로 표현하기 때문에 거리 근사의 오차가 커지는 단점이 있다. 이러한 단점을 극복하기 위하여, 본 저자들은 선행 연구에서 벡터간 각도 근사를 이용한 새로운 유클리드 거리 함수의 하한 함수를 제안한 바 있다[12]. 이 하한 함수는 두 벡터의 놈뿐만 아니라 두 벡터간 각도 성분을 고려함으로써 Cauchy-Schwartz 부등식을 적용한 하한 함수보다 거리 근사의 오차를 효과적으로 줄일 수 있다.

이 하한 함수를 사용한 정보검색은 벡터간 근사 거리에 의한 필터링 단계와 유클리드 거리를 이용하여 최종 정답을 판단하기 위한 후처리 단계로 구성된다. 이 때 거리 근사를 위해 저장하는 벡터는 차원 축소 기법에서 사용하는 일종의 저차원 벡터이다. 그러나 이 경우 고차원 데이터 벡터를 단지 각도와 놈의 두 개의 값으로 표현하기 때문에 원 특징벡터의 정보 손실이 커지는 단점이 존재한다. 결과적으로 필터링 단계의 계산 효율은 증가할 수 있으나, 착오 채택이 증가함으로써 전체 검색 성능을 떨어뜨리는 요인이 된다. 따라서 원 특징벡터의 정보 손실을 줄여 착오 채택의 수를 조절하고 전체 검색 성능을 높일 수 있는 차원 축소 기법이 요구된다. 따라서 본 논문에서는 각도 근사를 이용한 유클리드 거리의 하한 함수를 이용하는 효과적인 차원 축소 방법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 제 Ⅱ장에서는 관련 연구로서 Cauchy-Schwartz 부등식과 각도 성분을 고려한 유클리드 거리의 하한 함수를 살펴본다. 제 Ⅲ장에서는 제안하는 차원 축소 기법에 관하여 자세히 다룬다. 제 Ⅳ장에서는 다양한 성능 평가를 통하여 제안된 기법의 우수성을 검증한다. 끝으로, 제 Ⅴ장에서는 본 논문을 요약하고, 결론을 내린다.

Ⅱ. 관련 연구

본 장에서는 Cauchy-Schwartz 부등식과 각도 성분을 고려한 유클리드 거리의 하한 함수에 관한 기존 연구를 다룬다.

1. Cauchy-Schwartz 부등식을 이용한 유클리드 거리의 하한 함수

두 벡터 내적이란 동일 방향 성분에 대한 곱으로 정의된다. n차원 공간상의 두 벡터 X, Y에 대하여 벡터 내적을 $< X, Y >$로 표현할 때 정의식은 식 (1)과 같다.

$$< X, Y > = \sum_{i=1}^{n} x_i \cdot y_i \quad (1)$$

Cauchy-Schwartz 부등식은 벡터 내적과 벡터 놈(norm)과의 관계를 표현하는 것으로 식 (2)와 같이 벡터의 내적의 상한(upper bound)을 정의하고 있다.

$$< X, Y > \leq \|X\| \|Y\| \quad where \|X\|^2 = \sum_{i=1}^{n} x_i^2 \quad (2)$$

유클리드 거리 함수는 식 (3)과 같이 정의되며, 벡터 내적을 이용한 표현은 식 (4)와 같다.

$$D(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (3)$$

$$D(X, Y) = \sqrt{\|X\|^2 + \|Y\|^2 - 2 < X, Y >} \quad (4)$$

Cauchy-Schwartz 부등식을 사용한 유클리드 거리 근사 함수를 $D_{cs}(X, Y)$라고 정의하면 식 (5)와 같이 표현된다.

$$D_{cs}(X, Y) = \sqrt{\|X\|^2 + \|Y\|^2 - 2\|X\| \|Y\|} \quad (5)$$

식 (2)의 Cauchy-Schwartz 부등식의 정의로부터 유클리드 거리 함수 $D(X, Y)$와 $D_{cs}(X, Y)$와의 관계는 식 (6)과 같고, 따라서 $D_{cs}(X, Y)$는 유클리드 거리 함수 $D(X, Y)$의 하한(lower bound)이 된다.

$$D(X, Y) \geq D_{cs}(X, Y) \quad (6)$$

이러한 결과를 멀티미디어 정보검색에 적용하기 위해서 모든 데이터 벡터에 대하여 각 벡터의 놈을 저장해 둔다. 질의 벡터가 들어왔을 때 질의 벡터의 놈을 계산하고 저장되어 있는 각 데이터 벡터의 놈을 이용하여 두 벡터간의 유클리드 거리를 식 (5)를 이용하여 근사한다. 근사된 거리가 유사 허용치(ε)를 만족하는 경우 해당 데이터 벡터를 정답 후보 집합에 포함시킨다. 식 (6)으로부터 근사된 거리는 실제 거리보다 항상 작거나 같으며, 이 결과 착오 기각이 발생하지 않는다. 뿐만 아니라, 놈만을 이용하여 계산하므로 계산량이 차원 수에 비례하여 증가하는 유클리드 거리 계산에 비하여 계산 시간을 크게 줄일 수 있다.

2. 각도 성분을 고려한 유클리드 거리의 하한 함수

Cauchy-Schwartz 부등식을 이용한 유클리드 거리의 하한 함수 $D_{cs}(X, Y)$는 벡터가 이루는 각도 성분을 전혀 고려하고 있지 않다. 벡터의 놈만을 이용하기 때문에 계산의 효율성은 높지만 실제 유클리드 거리에 대한 근사 오차가 커지는 원인이 된다.

이러한 문제를 해결하기 위하여 선행 연구에서 각도 성분을 고려한 유클리드 거리 함수의 하한 함수를 제안한 바 있다[12]. 유클리드 거리 함수를 살펴보면, 식 (7)과 같이 두 벡터의 놈과 각도 성분을 알 경우 유클리드 거리를 구할 수 있다.

$$\begin{aligned} D(X, Y) &= \sqrt{\|X\|^2 + \|Y\|^2 - 2 < X, Y >} \\ &= \sqrt{\|X\|^2 + \|Y\|^2 - 2\|X\| \|Y\| cos\theta} \end{aligned} \quad (7)$$

질의 벡터와 데이터 벡터간의 내적 또는 각도 성분은 질의 벡터가 주어질 경우에만 구할 수 있다. 이미 질의 벡터가 주어진 상황에서 데이터 벡터와의 내적이 아닌 두 벡터간의 각도를 구하고 유클리드 거리를 계산하는 것은 오히려 계산량을 증가시키게 된다. 반면, 데이터 벡터의 놈만을 미리 저장해 놓고 Cauchy-Schwartz 부등식을 이용할 경우, 앞서 언급한 바와 같이 각도 성분을 무시하기 때문에 유클리드 거리에 대한 근사 오차가 커진다. 거리 계산에 있어서 각도 성분을 효과적으로 이용하기 위해서는 적은 연산만으로 질의 벡터와 데이터 벡터들 간의 각도 성분을 예측할 수 있는 정보가 필요하다. 이 때 필요한 정보는 질의 벡터가 없는 상황에서도 미리 구할 수 있어야 한다. 본 저자들은 선행 연구에서 질의 벡터와 데이터 벡터간의 각도 성분을 예측하기 위해서 그림 1과 같이 기준 벡터(reference vector) 개념을 새롭게 도입한 바 있다[12].

그림 1에서 보여지는 바와 같이 기준 벡터 R이 주어질 경우 기준 벡터와 데이터 벡터간의 각도 θ_{RX_i}를 미리 구하여 저장해 둘 수 있다. 질의 처리 과정에서 질의 벡터 Q가 주어지면 질의 벡터와 기준 벡터와의 각도 θ_{QR}를 먼저 구한다. 질의 벡터와 i번째 데이터 벡터 X_i간의 각도 성분 θ_{QX_i}는 식 (8)을 통해 근사할 수 있다. 각도 성분을 고려한 질의 벡터와 데이터 벡터간의 근사 거리 함수는 식 (9)로 주어진다.

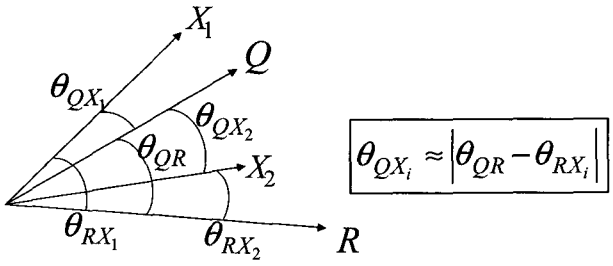


그림 1. 기준 벡터를 이용한 각도 근사 방법
Fig. 1. Angle approximation using a reference vector.

$$\widetilde{\theta_{QX_i}} = |\theta_{QR} - \theta_{RX_i}| \quad (8)$$

$$D_A(Q, X_i) = \sqrt{\|X_i\|^2 + \|Q\|^2 - 2\|X_i\|\|Q\|\cos\widetilde{\theta_{QX_i}}} \quad (9)$$

기준 벡터 R, 질의 벡터 Q, 그리고 각 데이터 벡터 X_i가 동일한 평면에 존재하지 않을 경우에도 θ_{QX_i}는 동일한 식에 의하여 근사할 수 있다. 이 때, 근사된 각도는 실제 각도보다 항상 작거나 같기 때문에 이에 기인한 착오 기각은 발생하지 않는다.

정리 1:

질의 벡터와 데이터 벡터 간의 실제 각도를 θ_{QX_i}, 각도 근사 방법에 의해 근사된 각도를 $\widetilde{\theta_{QX_i}}$라고 할 때 다음 식은 항상 성립한다.

$$\theta_{QX_i} \geq \widetilde{\theta_{QX_i}}$$

증명:

참고문헌 [12] 참조.

따름 정리 1:

임의의 두 벡터 X, Y에 대한 유클리드 거리 함수 $D(X, Y)$와 각도 근사를 이용한 거리 함수 $D_A(X, Y)$와의 관계에서 다음 식은 항상 성립한다.

$$D(X, Y) \geq D_A(X, Y)$$

증명:

참고문헌 [12] 참조.

각도 성분을 고려한 유클리드 거리의 하한 함수는 벡터의 놈과 기준 벡터와의 각도로 구성되는 두 성분만을 가지고 있다. 따라서 각도 근사를 이용한 하한 함수를 그대로 사용할 경우, 고차원 공간상의 특징 벡터를 놈과 각도 성분을 가지는 저차원 특징 벡터로 축소하는 효과를 가진다.

Ⅲ. 제안하는 기법

본 장에서 각도 성분을 고려하는 유클리드 거리의 하한 함수의 특징에 대하여 살펴보고, 각도 성분을 고려한 하한 함수에 기반한 효과적인 차원 축소 기법을 제안한다.

1. 각도 근사에 의한 오차 분석

기존 연구에서 제안했던 각도 근사 기법은 데이터 벡터 X, 질의 벡터 Q, 그리고 기준 벡터 R의 위치 관계에 따라 각도 근사 오차가 발생할 수 있다.

그림 2는 2차원 공간에서 세 벡터 X, Q, R의 예를 보여주고 있다. 그림에서와 같이 기준 벡터가 데이터 벡터와 질의 벡터 사이에 존재하지 않는 경우 각도 근사 기법에 의해 근사된 데이터 벡터와 근사 각도는 실제 각도와 일치한다. 그러나 기준 벡터가 데이터 벡터와 질의 벡터 사이에 존재하는 경우 근사 각도의 오차가 발생한다.

그림 3은 2차원 공간상에서 기준 벡터의 놈을 0.5로 고정해 놓고 기준 벡터의 각도를 x축을 기준으로 90도에서 10도 간격으로 0도까지 변화시키면서 데이터 벡터와 질의 벡터 사이의 각도를 추정하고, 추정한 각도와 실제 각도와의 차이를 나타낸 것이다. 여기서 x축을 기준으로 데이터 벡터 X는 약 66.8도, 질의 벡터 Q는 약

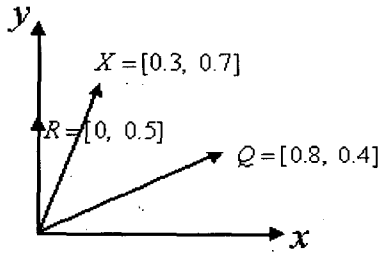


그림 2. 2차원 벡터 예시
Fig. 2. Example of 2-dimensional vectors.

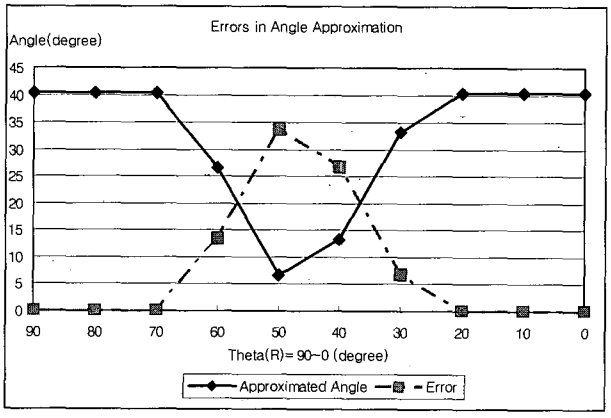


그림 3. 기준 벡터의 각도 변화에 따른 각도의 근사 오차
Fig. 3. Errors in angle approximation for varying angle of reference vector.

26.6도의 각도를 가진다. 그림에서 나타난 바와 같이, 기준 벡터가 데이터 벡터와 질의 벡터 사이에 존재할 때, 각도 근사의 오차가 발생함을 알 수 있다. 다수의 데이터 벡터들과의 근사 오차를 줄이기 위해서는 기준 벡터를 최대한 축에 가깝게 위치하도록 선정해야 한다.

그림 4는 3차원 공간에서 데이터 벡터 X, 질의 벡터 Q, 그리고 기준 벡터 R의 예를 보여주고 있다. 그림에서 나타난 바와 같이 세 벡터 X, Q, R은 z 차원 값이 모두 0.0이며, 이것은 세 벡터가 xy 평면상에 놓여있음을 의미한다. 세 벡터가 같은 평면상에 존재하고, R이 X와 Q 사이에 존재하지 않으므로, 각도 근사 기법에 의해 근사된 데이터 벡터와 질의 벡터 간의 각도는 실제 각도와 동일하다.

그림 5는 기준 벡터의 z 차원 값을 0에서 1까지 0.1씩 증가시키면서 추정한 각도와 실제 각도와의 차를 나타낸 것이다. 기준 벡터의 z 차원 값이 증가할수록 각도의 근사 오차는 커짐을 보여준다. 즉, 각도 근사에 사용되는 세 개의 벡터가 동일한 평면에 가까이 위치할수록 근사 각도의 오차는 적어진다.

그러나 차원의 수가 증가함에 따라 임의의 세 벡터가 동일한 평면 내에 존재할 확률은 낮아지기 때문에 고차원 공간에서는 근사 오차가 커지고 결과적으로 착오 채택이 많아지게 된다. 따라서 근사 오차를 줄이기 위해서는 각도 근사에 사용되는 벡터들이 하나의 평면에 가깝도록 하는 전략이 필요하다.

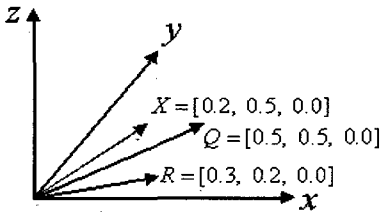


그림 4. 3차원 벡터 예시
Fig. 4. Example of 3-dimensional vectors.

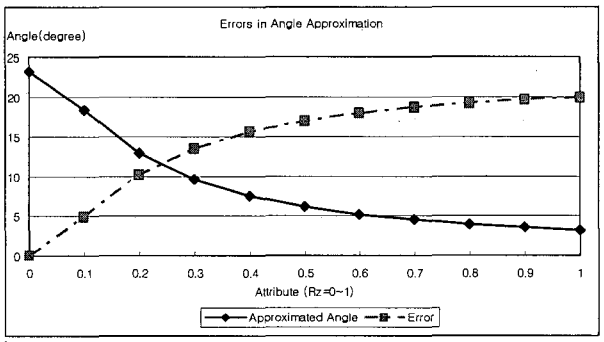


그림 5. 기준 벡터의 z 차원 값 변화에 따른 각도의 근사 오차
Fig. 5. Errors in angle approximation for varying attribute of z coordinate of reference vector.

데이터 벡터와 질의 벡터에 대하여 각도 근사 기법에 의해 근사된 각도의 오차를 줄이기 위해서, 기준 벡터는 첫째로 데이터 벡터, 질의 벡터와 동일한 평면에 가까워야 하고, 둘째로 최대한 축에 가깝게 위치하도록 선정되어야 한다.

2. 기본 전략

각도 근사를 이용한 유클리드 거리의 하한 함수 $D_A(X, Y)$를 사용하기 위해서는 원 데이터를 놈과 기준 벡터와의 각도 성분의 두 값만을 거리 근사를 위한 정보로 저장한다. 그러나 고차원 공간에서는 세 벡터가 하나의 평면에 가까울 확률이 매우 낮아 각도 근사 기법의 근사 오차가 커지는 단점이 있다. 반면, 저차원 공간에서 임의의 세 벡터가 하나의 평면에 가까워질 확률이 상대적으로 높다. 즉, 각도 근사 기법은 데이터 벡터의 차원이 낮아질수록 근사 각도의 오차가 작아지며, 이 결과 착오 채택의 수가 줄어든다.

본 연구에서는 이와 같은 특성을 반영하기 위하여 고차원 공간을 몇 개의 저차원 공간들의 집합으로 그루핑하고, 각 고차원 데이터 벡터를 몇 개의 저차원 데이터 벡터들로 간주하는 차원 그룹화(dimension grouping) 기법에 관하여 논의한다. 각 저차원 공간에서는 임의의 세 벡터가 하나의 평면에 가까워질 가능성이 상대적으로 높으므로 각도 근사 기법을 적용함으로써 전체 데이터 벡터에 대한 각도의 근사 오차를 줄일 수 있을 것이다. 또한, 데이터 벡터와 질의 벡터 사이에 기준 벡터가 존재하여 발생하는 근사 오차를 줄이기 위하여 축에 가까운 기준 벡터를 선정하는 방안에 대하여 논의한다.

3. 차원 그룹화를 이용한 차원 축소

본 논문에서는 n개의 차원들을 k개의 그룹으로 묶는 경우 차원 축소를 위한 그룹 수 k를 축소 차원 수(reduced dimensionality)라고 정의한다. 또한, 축소 차원 수 k를 저차원 특징 벡터의 차원과 동일한 의미로 사용한다. n차원 데이터 벡터 X의 전체 차원 중 일부 차원 값들만을 갖는 벡터를 X의 서브 벡터(subvector)라 정의한다. X_i'을 데이터 벡터 X의 i번째 서브 벡터라고 하고 $X_i' = [x_{i1}, x_{i2}, \cdots, x_{il_i}]$로 표현할 때, k개의 서브 벡터들은 식 (10)과 식 (11)의 두 조건을 만족한

다. 여기서 $A(X)=\{\forall x_i | x_i \in X\}$, 즉 벡터 X의 개별 차원으로 구성된 집합이라 정의한다.

$$\{A(X_i') \cap A(X_j') | \forall i,j(\leq k), i \neq j\} = \varnothing \quad (10)$$

$$A(X) = \bigcup_{i=1}^{k} A(X_i') \quad (11)$$

l_i는 i번째 서브 벡터의 차원 값의 개수를 나타내고 $\sum_{i=1}^{k} l_i = n$이 된다. 기준 벡터 R에 대해서도 동일하게 k개의 서브 벡터 R_i'로 표현이 가능하다. 각 서브 벡터의 놈과 기준 서브 벡터와의 각도 성분은 각각 식 (12)와 식 (13)과 같이 구할 수 있다. 이 때 $X_i' = [x_{i1}, x_{i2}, \cdots, x_{il_i}]$, $R_i' = [r_{i1}, r_{i2}, \cdots, r_{il_i}]$ 이다.

$$\|X_i'\| = \sqrt{\sum_{j=1}^{l_i} x_{ij}^2} \quad (12)$$

$$\theta_{X_i}' = cos^{-1}\left(\sum_{j=1}^{l_i} x_{ij} r_{ij} / \|X_i'\| \|R_i'\|\right) \quad (13)$$

따라서 X를 k개의 서브 벡터로 분리하는 경우, X의 차원 축소된 데이터 벡터 X_{GA}는 식 (14)와 같이 표현된다. 또한, 두 개의 축소된 차원 내 저차원 데이터 벡터 X_{GA}, Y_{GA}간 거리 $D_{GA}(X, Y)$를 식 (15)와 같이 정의한다.

$$X_{GA} = [\|X_1'\|, \theta_{X_1}', \|X_2'\|, \theta_{X_2}', \cdots, \|X_k'\|, \theta_{X_k}'] \quad (14)$$

$$D_{GA}(X, Y) = \sqrt{\sum_{i=1}^{k} (\|X_i'\|^2 + \|Y_i'\|^2 - 2\|X_i'\| \|Y_i'\| \cos\widetilde{\theta_{X_i'Y_i'}})}$$
$$where\ \widetilde{\theta_{X_i'Y_i'}} = |\theta_{X_i}' - \theta_{Y_i}'| \quad (15)$$

식 (14)는 제안하는 차원 축소 기법에 의해 고차원 데이터 벡터를 변환한 축소 차원 수 k인 축소된 데이터 벡터를 나타낸다. 이 때, $D_{GA}(X, Y)$는 필터링 단계에서 유클리드 거리를 대신하여 사용되는 거리 함수가 된다.

4. 기준 벡터 선정

앞서 제 3.1절에서 살펴본 바와 같이, 각도 근사 오차를 줄이기 위해서는 전체 데이터 벡터에 대하여 질의 벡터와 기준 벡터가 이루는 평면과의 거리가 최소화 되도록 기준 벡터를 설정할 필요가 있다. 기준 벡터가 데이터 벡터와 질의 벡터 사이에 존재함으로써 발생하는 각도 근사의 오차를 줄이기 위해서는 최대한 구석에 존재하는 기준 벡터를 선정할 필요가 있다. 이 때 데이터 공간상의 구석에 존재하는 벡터의 의미는 데이터 공간을 형성하는 축에 가까운 데이터 벡터를 말한다.

일반적으로 질의 벡터의 분포는 데이터 벡터와 유사한 분포를 따른다고 가정한다[21]. 따라서 기준 벡터, 데이터 벡터, 그리고 질의 벡터가 하나의 평면에 가깝도록 하기 위해서는 가능한 전체 데이터 벡터들이 이루는 평면까지의 거리가 최소화되도록 기준 벡터를 설정하는 방안이 필요하다. 본 연구에서는 이를 위하여 Levenberg-Marquardt(L-M) 알고리즘을 이용한 기준 벡터 설정 방법을 제안한다.

n차원 평면의 법선 벡터를 $V = [v_1, v_2, \cdots v_n]$라고 할 때, 평면의 방정식은 식 (16)과 같다. 여기서 데이터 벡터와 축에 대한 표기상의 혼동을 피하기 위하여 n차원 축을 $A = \{a_1, a_2, \cdots, a_n\}$로 표기한다. 이 평면과 i번째 데이터 벡터 X_i에 대하여 식 (17)을 만족하는 상수 T_i가 존재하고, 법선 벡터 V의 크기가 1이라고 할 때 상수 T_i는 X_i와 평면과의 거리가 된다. 식 (17)을 T_i에 대하여 정리하면 식 (18)과 같은 거리 함수를 얻을 수 있다.

$$v_1 a_1 + v_2 a_2 + \cdots v_n a_n = 0 \quad (16)$$

$$v_1(x_{i1} + T_i v_1) + v_2(x_{i2} + T_i v_2) + \cdots + v_n(x_{in} + T_i v_n) = 0 \quad (17)$$

$$T_i = -\frac{v_1 x_{i1} + v_2 x_{i2} + \cdots + v_n x_{in}}{v_1^2 + v_2^2 + \cdots + v_n^2} \quad (18)$$

법선 벡터 V의 크기가 1이라고 할 때, N개의 모든 데이터 벡터로부터 평면까지의 제곱 거리는 식 (19)과 같다. 따라서 식 (19)을 최소화하는 법선 벡터 V를 구하면 이 평면은 모든 데이터 벡터로부터의 거리가 최소가 되는 평면이 된다.

$$\sum_{i=1}^{N} T_i^2 = \sum_{i=1}^{N} (v_1 x_{i1} + v_2 x_{i2} + \cdots v_n x_{in})^2 \quad (19)$$

본 논문에서는 Levenberg-Marquardt 최소화 알고리즘을 사용하여 식 (19)를 최소화하는 법선 벡터 V를 구하였다. 다음으로 이 평면에 대하여 모든 데이터 벡터의 정사영을 구하고, 각도 차이가 가장 큰 한 쌍의 정사영 중 하나를 기준 벡터로 선정한다. 이는 선정된 기준 벡터가 모든 데이터 벡터와 같은 평면에 존재할 가능성을 높일 뿐만 아니라 데이터 벡터와 질의 벡터 사이에 존재하여 발생하는 근사 오차도 줄일 수 있다.

기준 벡터의 모든 차원 값을 매우 작은 값으로 줄 경우, 이 기준 벡터는 모든 축에 가까운 벡터가 된다. 따라서 이 경우에는 데이터 벡터와 질의 벡터의 사이에 기준 벡터가 존재할 가능성이 매우 낮다. 극단적으로 기준 벡터의 모든 차원 값을 -1로 주면, 데이터 벡터 공간에서 벗어나지만 데이터 벡터와 질의 벡터의 사이에 존재할 가능성은 전혀 없다.

제안하는 차원 축소 기법이 기준 벡터 선정에 따른 영향을 살펴보기 위해서, 제 4장에서 다양한 기준 벡터에 대한 실험 결과를 제시하고 성능을 비교 평가한다.

5. 데이터베이스 구축 및 질의 처리 과정

제 3.4절에서 제시한 방식으로 데이터 벡터들을 분석하여 기준 벡터를 선정한다. 기준 벡터가 선정되면, 모든 데이터 벡터에 대하여 식 (12)와 식 (13)을 이용하여 축소 차원 수 k인 저차원 데이터 벡터를 구하고, 이를 데이터베이스에 저장해 둔다. 기존의 PCA와 DCT를 이용한 차원 축소 기법과 제안하는 기법은 축소 차원 수를 조절할 수 있다. 축소된 저차원 데이터 벡터의 저장 공간을 비교해 보면, 각도 성분의 경우 정수 값으로 $[0,\pi]$구간의 값을 표현한다면 1바이트가 필요하다. 따라서 PCA와 DCT를 이용한 차원 축소에서 차원 축소된 차원 값을 4바이트로 표현할 때, 제안하는 기법은 놈을 4바이트, 각도 성분을 1바이트로 표현하기 때문에 25%의 저장 공간이 더 필요하다.

질의 처리 과정에서는 질의 벡터가 주어지기 이전에 구축된 저차원 데이터 벡터들의 집합과 원 데이터 벡터들의 집합이 저장된 데이터베이스를 이용한다. 질의 벡터가 주어졌을 때, 필터링 단계에서는 저차원 데이터 벡터들을 이용하고, 후처리 단계에서는 원 데이터 벡터들의 집합을 이용한다.

질의 처리 알고리즘

1. 질의 벡터에 대한 저차원 벡터 Q_{GA}를 생성한다.
2. 모든 저차원 데이터 벡터 X_i에 대하여,
   - 2.1. $D_{GA}(X_i, Q)$를 계산한다.
   - 2.2. 거리값이 ε보다 작을 경우 해당 데이터 벡터를 후보 집합에 포함시킨다.
3. 후보 집합에 포함된 데이터 벡터 X_i에 대하여,
   - 3.1. $D(X_i, Q)$를 계산한다.
   - 3.2. 거리값이 ε보다 작을 경우 정답으로 반환한다.

질의 벡터에 대한 저차원 벡터 Q_{GA}는 식 (12)와 식 (13)을 통해 생성된다. 필터링 단계에서 사용자 정의 유사 허용치 ε과 비교하기 위한 거리값은 식 (15)를 이용하여 계산한다. 이 때, 모든 데이터 벡터에 대한 저차원 벡터는 이미 데이터베이스로 구축되어 있기 때문에 다시 계산할 필요가 없으며, 질의 벡터와 데이터 벡터간의 각도 성분은 단순 빼기 연산만으로 근사되기 때문에 계산량은 크지 않다. 후처리 단계에서는 고차원 데이터 벡터의 모든 차원을 사용하여 정답 여부를 결정한다.

6. 논의 사항

제 2.2절의 정리 1을 통해 이미 $D_A(X, Y)$를 이용한 검색에서는 정답이 필터링 단계 후 항상 후보에 포함되기 때문에 착오 기각이 발생하지 않음을 보였다. 본 절에서는 정리 2를 통하여 제안한 차원 축소 기법이 착오 기각을 발생시키지 않음을 보인다.

정리 2:

임의의 두 벡터 X, Y에 대하여 다음 식은 항상 성립한다.

$$D(X, Y) \geq D_{GA}(X, Y)$$

증명:

두 벡터 X, Y에 대하여 서브 벡터 $X_i' = [x_{i1}, x_{i2}, \cdots, x_{il_i}]$이고, $Y_i' = [y_{i1}, y_{i2}, \cdots, y_{il_i}]$ 이다. 두 서브 벡터간의 유클리드 거리 $D(X_i', Y_i')$는 식 (20)과 같다.

$$D(X_i', Y_i') = \sqrt{\|X_i'\|^2 + \|Y_i'\|^2 - 2 < X_i', Y_i' >} \quad (20)$$

같은 방식으로 두 서브 벡터에 대하여 각도 근사를 이용한 거리 $D_A(X_i', Y_i')$는 식 (21)과 같다.

$$D_A(X_i', Y_i') = \sqrt{\|X_i'\|^2 + \|Y_i'\|^2 - 2\|X_i'\|\|Y_i'\|\cos\theta_{\widehat{X_i'Y_i'}}} \quad (21)$$

두 벡터 X, Y에 대하여, 유클리드 거리와 서브 벡터에 대한 유클리드 거리와의 관계식을 정리해 보면 식 (22)와 같다. 또한, 식 (15)와 식 (21)로부터 $D_{GA}(X, Y)$와 서브 벡터들 간의 거리 관계는 식 (23)과 같이 정리할 수 있다.

$$D(X, Y)^2 = \sum_{i=1}^{k} D(X_i', Y_i')^2 \quad (22)$$

$$D_{GA}(X, Y)^2 = \sum_{i=1}^{k} D_A(X_i', Y_i')^2 \quad (23)$$

정리 1로부터 $D(X_i', Y_i') \geq D_A(X_i', Y_i')$은 항상 성립한다. 따라서 식 (22)와 식 (23)으로부터 $D(X, Y) \geq D_{GA}(X, Y)$가 항상 성립한다. 즉, 그룹화 기법을 적용한 거리 함수는 유클리드 거리 함수의 하한 함수이다. □

제안하는 기법에서 사용하는 거리 함수는 유클리드 거리 함수의 하한 함수임을 정리 2에서 보였다. 즉, 축소된 데이터 벡터 공간에서의 두 벡터간의 거리 $D_{GA}(X, Y)$가 원 데이터 벡터 공간에서 벡터간의 거리 $D(X, Y)$ 보다 항상 작거나 같아야 한다. 따라서 D_{GA}를 필터링 단계에서 사용하는 제안된 기법에서는 질의 처리 시 착오 기각이 발생하지 않는다.

제안하는 차원 축소 기법은 축소 차원 수를 사전에 정의함으로써 저차원 데이터 벡터의 차원에 대한 조절이 가능하다. 기존에 제안된 놈과 각도 성분을 단독으로 사용할 때, 근사된 각도에 오차가 있는 경우 데이터 벡터와 질의 벡터와의 유클리드 거리를 하나의 놈과 오차를 포함하는 각도 성분으로 근사하기 때문에 전체 차원에 영향을 주게 된다. 이에 반하여 저차원 데이터 벡터로 그룹을 생성하게 되면 각도 근사 오차를 줄일 수 있을 뿐만 아니라 오차가 각 그룹 내부에만 영향을 주기 때문에 필터링 단계에서 유클리드 거리에 대한 근사 오차가 작아진다. 결과적으로 착오 채택되는 후보 개수도 효과적으로 줄일 수 있다.

Ⅳ. 성능 평가

본 장에서는 제안하는 차원 축소 기법의 성능을 평가하기 위한 실험 환경과 실험 결과를 제시한다.

1. 실험 환경

본 논문에서는 성능을 평가하기 위해 합성 데이터와 실제 데이터인 Corel 영상 데이터[26]를 사용하였다. 합성 데이터는 25차원에서 200차원까지 다양한 차원 수를 갖는 벡터들의 집합이다. 각 데이터 집합은 클러스터들의 집합으로 구성되며, 20,000개에서 100,000개까지 다양한 수의 데이터들을 포함한다.

각 클러스터 내에 속하는 벡터들은 다음과 같은 방식을 이용하여 생성된다. (1) 클러스터 내에 속하게 될 데이터 수를 5에서 50 사이에서 무작위로 선택한다. (2) 클러스터의 축 시스템(axis system)을 무작위로 결정한다. (3) 각 축에 대하여 데이터들이 평균값 0, 표준편차값이 1,000,000~100,000,000 사이인 정규 분포를 취하도록 데이터의 각 축 값을 생성한다. (4) 생성된 클러스터 내의 데이터들을 단계 (1)에서 결정한 축 시스템에 맞도록 회전시킨다. (5) 각 축에 대하여 클러스터의 중심점의 값을 -2,000,000,000 ~ 2,000,000,000 사이에서 무작위로 선택한 후, 클러스터 내의 데이터들을 선택된 중심점에 맞도록 이동시킨다. 최종적으로 각 차원 값으로 [0, 1]의 구간 내의 실수 값을 갖도록 정규화 시킨다.

Corel 영상 데이터는 총 68,040장의 영상으로 구성되며, 각 영상은 32차원의 특징 벡터로 표현된다. 이 특징 벡터내의 각 차원 값 역시 [0, 1] 구간에 속하는 값을 갖는다. 그림 6은 실험을 위해 생성한 200차원의 합성

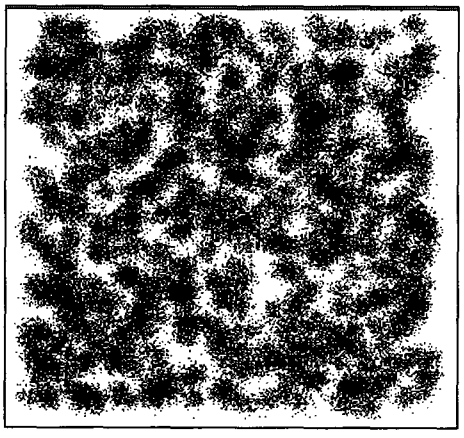


그림 6. 2차원으로 사상한 데이터 분포
Fig. 6. Experimental data distribution projected to 2-dimension.

데이터를 2차원으로 사상(projection)하여 도면화한 것이다. 그림에서와 같이 합성 데이터는 다수의 클러스터를 이루고 있으며 클러스터 내부는 데이터 벡터 사이 구별이 힘들만큼 밀집되어 있다.

제안하는 차원 축소 기법의 성능을 기존의 PCA를 이용한 차원 축소 기법과 DCT를 이용한 차원 축소 기법과 비교하였다. 하나의 놈과 각도를 사용하는 경우는 제안하는 차원 축소 기법에서 $k=1$로 고정된 경우를 의미한다. 질의 처리 성능을 평가하기 위하여 후보 개수를 비교하였다. 이는 각 기법에서 착오 채택이 얼마나 많이 발생하는가를 비교하기 위한 척도로서, 후보 개수가 적을수록 검색 성능이 좋다고 할 수 있다. 본 실험에서는 총 100개의 임의의 질의 벡터에 대하여 후보 개수를 각각 구하고 이들의 평균을 취하였다. 또한 축소된 차원 수 k의 변화에 따른 평균 후보 개수의 변화를 비교하였다.

본 실험에서는 주기억장치에 데이터 벡터를 저장하였다. 성능 평가를 위한 하드웨어 플랫폼은 2.8G Pentium IV와 512MB의 주기억장치가 장착된 PC이며, 소프트웨어 플랫폼은 MS Windows 2000 및 Visual C++6.0이다.

2. 실험 결과

실험에 대한 구현의 정확함을 검증하기 위해 원 데이터 벡터의 모든 차원에 대한 유클리드 거리를 사용한 상세 검색의 최종 결과와 PCA를 이용한 차원 축소 기법, DCT를 이용한 차원 축소 기법, 그리고 본 논문에서 제안하는 차원 축소 기법 등을 이용한 2단계 검색의 최종 결과를 비교하였다. 검증 결과, 세 가지 기법 모두 검색 결과 착오 기각 없이 상세 검색과 동일한 검색 결과를 나타냄을 확인하였다.

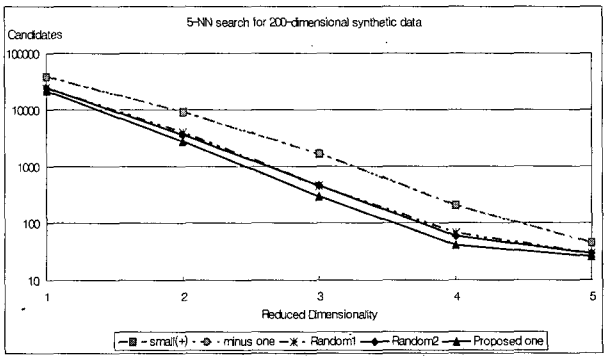


그림 7. 기준 벡터에 따른 성능 비교(로그 스케일)
Fig. 7. Experimental result according to reference vectors(logarithmic scale).

제안하는 차원 축소 기법은 각도 근사를 위해 기준 벡터가 필요하다. 이 때, 어떤 기준 벡터를 사용하느냐에 따라 성능의 차이가 발생할 수 있다. 다양한 기준 벡터에 대한 성능을 비교하기 위하여 k의 변화에 따른 필터링 단계 후의 후보 개수를 비교하였다. 필터링 단계에서 사용한 거리 함수는 $D_{GA}(X, Y)$이다.

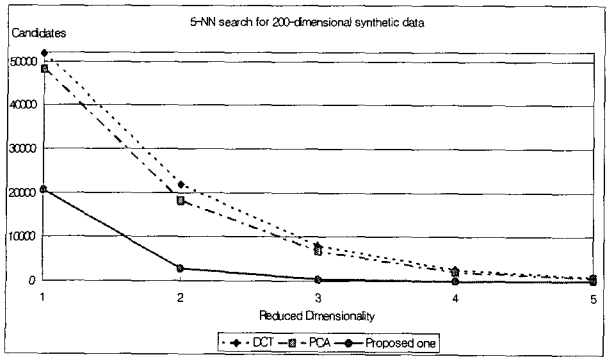
그림 7은 본 실험에 대한 결과를 나타낸 것이다. 축소 차원 수가 증가하면 후보 개수가 급격히 줄기 때문에 일반 스케일로 표현할 경우 상대적인 비교가 어렵다. 그림 7에서는 상대적인 비교를 쉽게 할 수 있도록 하기 위하여 세로축을 후보 개수에 대한 로그 스케일로 표현하였다. 향후 실험에서는 상대적인 비율의 비교를 쉽게 할 수 있도록 세로축에 대하여 일반 스케일과 로그 스케일 모두를 보였다.

small(+)는 축에 가까운 벡터를 기준 벡터로 선정하기 위해 모든 차원 값을 매우 작은 값으로 준 것이다. minus one은 모든 차원 값을 -1로 준 경우이다. Random1과 Random2는 데이터 벡터 중 임의의 두 벡터를 기준 벡터로 선정한 경우이다. 마지막으로 Proposed one은 본 연구에서 제안한 기준 벡터 선정 방법에 의해 선정된 기준 벡터를 사용한 경우이다.

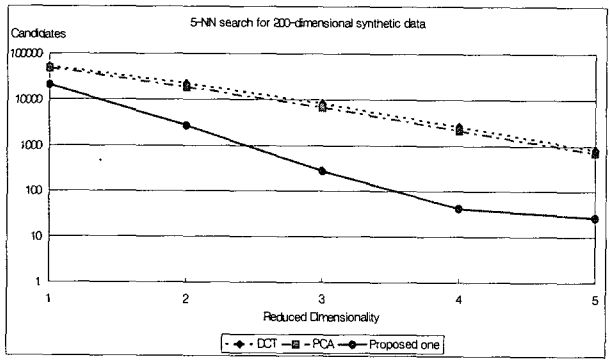
결과에서 보는 바와 같이 제안한 기준 벡터 선정 방법의 경우가 가장 좋은 성능을 보이는 것으로 나타났다. 제안하는 방법은 제 3.1절에서 언급한 근사 오차의 원인을 모두 감안하여 기준 벡터를 선정한 것이다. 축에 가깝게 존재하는 기준 벡터인 small(+)와 minus one의 결과를 보면, 각도의 근사 오차 원인 중 세 벡터가 하나의 평면에서 멀어짐으로써 발생하는 오차가 더 지배적인 영향을 줌을 알 수 있다. 이후 모든 실험에서는 제안하는 방법에 의해 선정한 기준 벡터를 사용한다.

축소된 차원이 커질수록 원 데이터 벡터의 정보 손실은 줄어든다. 따라서 필터링 단계 이후의 후보 개수도 줄어든다. 이에 대한 성능을 비교 평가하기 위하여 축소된 차원의 변화에 따른 후보 개수의 변화에 대한 실험을 수행하였다.

그림 8은 200차원의 데이터 벡터 100,000개에 대하여 k의 변화에 따른 필터링 후의 후보 개수를 보여주고 있다. 그림 8-(a)는 세로축을 후보 개수를 일반 스케일로 표현한 결과를 보여주고 있다. 그림 8-(a)에서 축소 차원 수가 4 또는 5일 경우 거의 비슷한 성능을 나타내는 것처럼 보이지만, 실제 비율을 비교해 보면 오히려 큰 성능 차이가 난다. 이러한 비율을 쉽게 비교할 수 있도



(a) 일반 스케일
(a) Original scale.



(b) 로그 스케일
(b) Logarithmic scale.

그림 8. 축소 차원 수의 변화에 따른 성능 비교
Fig. 8. Experimental result according to varying reduced dimensionality.

록 하기 위하여 그림 8-(b)에서 후보 개수에 대한 로그 스케일로 세로축으로 표현하였다.

기존의 PCA나 DCT를 이용한 차원 축소 기법과 제안하는 차원 축소 기법에서 모두 축소 차원 수가 증가할수록 필터링 단계 후의 후보 개수가 현저히 작아짐을 볼 수 있다. 또한, 모든 경우에서 제안한 기법이 기존의 두 기법보다 나은 성능을 보임을 확인할 수 있다. 전술한 바와 같이 실험을 위해 생성한 데이터는 다수의 클러스터를 보유하고 있으며, 각 클러스터 내에 여러 데이터 벡터가 포함되도록 구성하였다. 클러스터 내의 데이터 벡터 수는 매우 작기 때문에 원 데이터 벡터의 모든 차원을 사용하기 이전에는 서로 구별하기 어려울 정도로 유사하다[5].

본 실험의 경우, 제안하는 기법은 축소 차원 수가 4에서 5로 증가할 때 낮은 축소 차원 수에 비해 성능 향상 정도가 작아지는 현상을 보인다. 이는 축소 차원 수가 4 이상일 경우, 이미 클러스터간의 구별을 대부분 할 수 있음을 의미한다. 실제 축소 차원 수를 5 이상으로 늘렸을 때 5인 경우와 동일한 수의 후보가 선택됨을 확인하였다. 이에 비하여 PCA나 DCT를 이용한 차원 축소 기법을 이용한 경우는 축소 차원 수가 5인 경우에도 아직 클러스터들 간의 구별이 정확히 되지 않음을 보이고 있다. 후보 개수를 비교하면, 제안하는 기법은 DCT를 이용한 기법과 비교하여 약 2배에서 60배까지 성능이 향상되었고, PCA를 이용한 기법과 비교해서는 약 2배에서 50배까지 성능이 향상됨을 확인하였다.

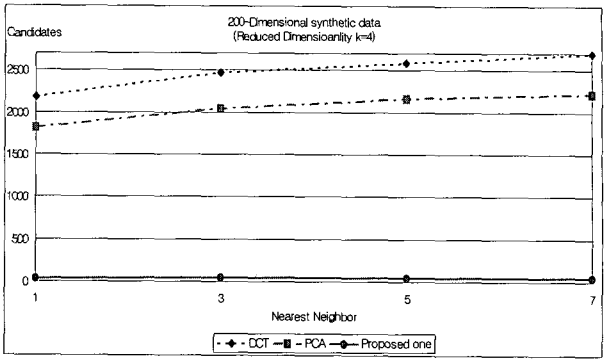


그림 9. 유사 허용치 변화에 따른 성능 비교
Fig 9. Experimental result according to varying tolerance.

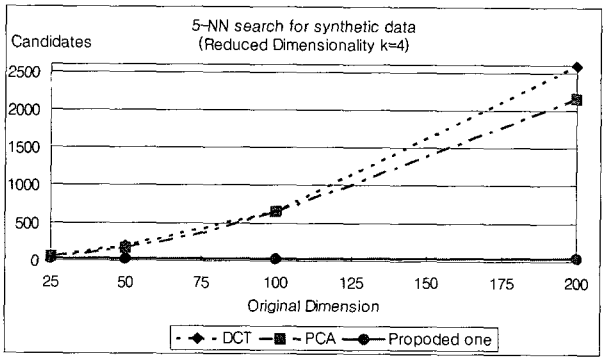


그림 10. 데이터 벡터의 원본 차원에 따른 성능 비교
Fig. 10. Experimental result according to varying dimensionality.

유사 허용치가 증가하면 후보 개수는 증가하게 된다. 이러한 경향을 살펴보기 위하여 유사 허용치 ε의 변화에 따른 세 기법의 질의 처리 성능을 비교하였다. 유사 허용치는 1-NN에서부터 7-NN까지 선택될 수 있는 범위를 사전 실험을 통해 결정한 후, 본 실험의 범위 질의를 수행하였다. 원 데이터 벡터는 200차원이고, 데이터 개수는 100,000개로 구성되어 있으며, k는 4로 고정하였다.

그림 9는 유사 허용치 ε의 변화에 따른 실험 결과를 보여주고 있다. 세 가지 방법 모두 유사 허용치가 커짐에 따라 후보 개수가 증가함을 알 수 있다. PCA와 DCT를 이용한 차원 축소 기법에서는 ε이 증가함에 따

라 후보 개수가 크게 증가하는 반면, 제안하는 기법은 거의 비슷한 수준을 보임을 알 수 있다. 제안하는 기법은 DCT를 이용한 기법의 약 60배, PCA를 이용한 기법의 약 50배의 성능이 향상이 있음을 확인하였다.

데이터 벡터의 차원에 따른 성능을 평가하기 위하여 데이터 벡터의 차원을 25에서 200까지 변화시킨 합성 데이터를 대상으로 5-NN 검색 실험을 수행하였다. 사용된 합성 데이터의 수는 100,000개이며, k는 4로 고정하였다. 그림 10은 본 실험에 대한 결과를 나타낸 것이다.

그림 10에서 보는 바와 같이, 기존의 PCA나 DCT를 이용한 기법의 경우 원 데이터 벡터의 차원 수가 증가함에 따라 후보 개수가 급격하게 증가하는 경향을 보인다. 제안하는 기법의 경우는 원 데이터 벡터의 차원이 증가하더라도 후보 개수가 거의 비슷한 수준으로 유지됨을 알 수 있다. 제안하는 기법은 원 데이터 벡터가 25차원인 저차원에서는 DCT와 PCA를 이용한 기법과 비교하여 약 2배 정도의 성능 향상을 보였으나, 고차원인 200차원에서는 DCT의 약 60배, PCA의 약 50배의 성능 향상이 있음을 확인하였다. 이는 제안하는 기법이 특징 벡터가 고차원화 되어 가는 멀티미디어 정보 검색에 매우 적합하다는 것을 보이는 것이다.

다음으로, 데이터 벡터의 개수 증가에 따른 검색 성능을 비교하였다. 데이터 벡터의 차원은 200차원으로 고정하고, 5-NN 검색 실험을 진행하였다. 본 실험에서도 k는 동일하게 4로 고정하였다. 그림 11은 데이터 벡터의 개수에 따른 성능을 나타낸 것이다.

데이터 벡터의 개수가 증가하는 경우에도 기존의 PCA나 DCT를 이용한 기법의 경우 후보 개수가 크게 증가함을 알 수 있다. 이는 고차원 공간에서 데이터 벡터가 넓게 분포하게 되는 통계적인 특성에 기인한 것이다. 제안하는 기법의 경우 데이터 벡터의 개수가 증가하더라도 후보 개수가 약간씩 증가하지만 거의 비슷한 수준을 유지함을 알 수 있다. 또한, DCT를 이용한 기법과 비교하여 약 18배에서 최대 약 60배, PCA를 이용한 기법과 비교해서는 약 15배에서 최대 약 50배의 성능 향상을 보였다. 이 역시 제안하는 기법이 대용량화 되어가는 멀티미디어 정보 검색에 있어 적합하다는 것을 보이는 결과이다.

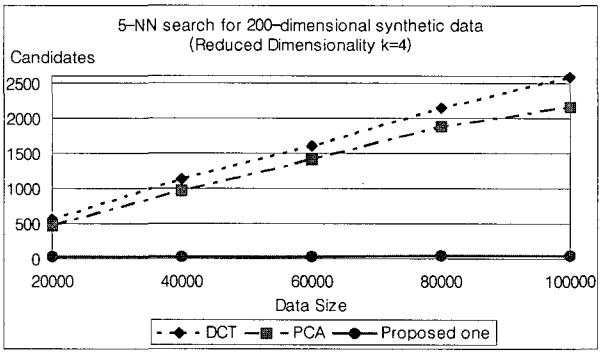
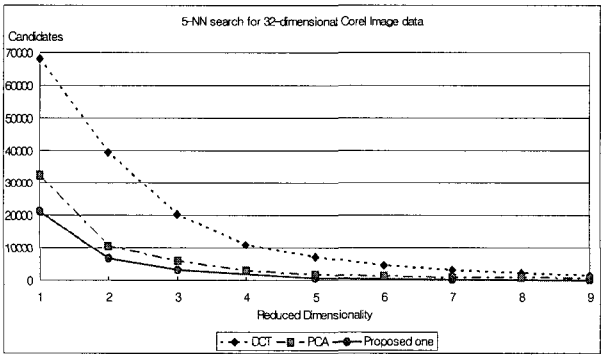


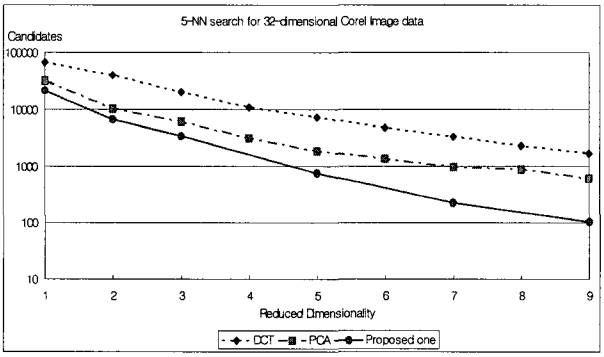
그림 11. 데이터 벡터의 개수에 따른 성능 비교
Fig 11. Experimental result according to the number of data vectors.

마지막으로, 실제 Corel 영상 데이터를 사용하여 세 기법의 성능을 비교하였다. 그림 12는 실제 데이터인 Corel 영상 데이터에 대하여 축소된 차원 값에 따르는 후보 개수의 변화를 보여주고 있다.

$k=1$인 경우를 비교해 보면, DCT를 이용한 기법은 모든 데이터를 후보로 선택하고 있다. 반면 PCA를 이용한 기법은 47.6%, 제안하는 차원 축소 기법은 31.1%만을 후보로 선택하고 있다. 다시 말해서 하나의 정보만을 비교하여 제안된 기법은 정답 가능성이 전혀 없는 68%이상의 데이터를 제거할 수 있었다. $k=9$의 경우,



(a) 일반 스케일
(a) Original scale.



(b) 로그 스케일
(b) Logarithmic scale.

그림 12. Corel 영상 데이터에 대한 k의 변화에 따른 성능 비교
Fig 12. Experimental result according to varying k for Corel image data.

DCT를 이용한 기법은 2.47%만을 후보로 선정하였으며, PCA를 이용한 기법은 0.89%만을 선택하였다. 제안하는 기법은 0.15%만을 후보로 선택하였다. PCA나 DCT를 이용한 기법들과 비교하여 제안된 기법은 각각 약 6배와 약 16배의 성능 향상 효과를 보였다. 합성 데이터에서 최대 성능 향상 정도를 보이는 200차원에 비해 성능 향상의 정도가 상대적으로 낮아 보이지만, 32차원의 합성 데이터와 비교해 보면, Corel 영상 데이터에 대한 실험에서의 성능 향상 정도가 더 높다.

V. 결 론

멀티미디어 정보검색에서 멀티미디어 데이터는 고차원 공간의 벡터로 표현되며, 검색을 위한 유사도 기준으로는 벡터간의 유클리드 거리가 널리 사용되고 있다. 고차원 데이터 벡터에 대하여 색인 구조를 적용할 경우, 검색 성능이 급격히 떨어지는 차원의 저주 문제가 발생한다. 차원의 저주 문제를 해결하기 위한 하나의 접근 방법은 고차원 데이터 벡터를 저차원 공간으로 변환하여 저차원 벡터를 색인하는 것이다.

본 연구에서는 유클리드 거리의 하한 함수를 기반으로 하는 차원 축소 기법을 제안하였다. 사용한 하한 함수에서는 기준 벡터라 부르는 별도의 벡터를 이용하여 추정된 두 벡터간의 각도 성분을 그들을 위한 유클리드 거리 근사에 사용한다. 그러나 이 하한 함수는 고차원 데이터를 단지 놈과 각도 성분의 두 개의 값으로만 표현하기 때문에 각도 근사 오차가 증가하고 유클리드 거리 근사에 많은 영향을 주는 문제를 가지고 있다. 이를 해결하기 위해 고차원 데이터를 저차원 서브 벡터의 그룹으로 표현함으로써 축소된 차원의 저차원 벡터로 표현하는 차원 축소 기법을 제안하였다. 제안한 차원 축소 기법에서 축소된 저차원 공간상의 거리 함수 역시 유클리드 거리의 하한 함수가 됨을 이론적으로 증명하였다. 또한, 합성 데이터와 실제 데이터를 이용한 다양한 실험을 통하여 제안하는 방법의 우수성을 규명하였다.

합성 데이터에 대한 실험 결과에 의하면, 제안된 차원 축소 기법은 필터링 단계 후 후보 개수를 비교해 볼 때, DCT를 이용한 기법과 비교하여 2배에서 60배, PCA를 이용한 기법과 비교하여 2배에서 50배의 성능 향상 효과가 있음을 보였다. Corel 영상 데이터에 대해서는 PCA나 DCT를 이용한 기법들과 비교하여 각각 약 6배와 16배의 성능 향상 효과가 있음을 보였다. 특히, 제안하는 기법은 차원 수나 데이터 개수의 증가에 따르는 급격한 성능 저하 현상을 보이지 않는다. 이는 나날이 고차원화 대용량화 되어가는 멀티미디어 정보검색 분야의 특성을 고려할 때, 매우 바람직한 결과라 할 수 있다.

향후 연구를 통해 고차원 데이터 벡터를 서브 벡터의 그룹으로 나눌 경우 각 그룹에 속하는 속성의 최적 개수와 어떤 속성들 끼리 그룹을 지을 것인가에 대한 최적화 방안을 고안할 예정이다.

참 고 문 헌

[1] C. C. Aggarwal, "On the Effects of Dimensionality Reduction on High Dimensional Similarity Search," In *Proc. Int'l. Symp. on Principles of Database Systems, ACM SIGACT-SIGMOD-SIGART*, pp. 256-266, May 2001.

[2] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search in Sequence Database," In *Proc. Int'l. Conf. on Foundations of Data Organization and Algorithms, FODO*, pp. 69-84, Oct. 1993.

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

[4] S. Berchtold, C. Böhm, B. Braunmüller, D. Keim, and H.-P. Kriegel, "Fast Parallel Similarity Search in Multimedia Databases," In *Proc. Int'l. Conf. on Management of Data, ACM SIGMOD*, pp. 1-12, 1997.

[5] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is Nearest Neighbor Meaningful?," In *Proc. Int'l. Conf. on Database Theory, IDCT*, pp. 217-235, Jan. 1999.

[6] C. Böhm, S. Berchtold, and D. Keim, "Searching in High-Dimensional Spaces-Index Structures for Improving the Performance of Multimedia Databases," *ACM Computing Surveys*, Vol. 33, Issue 3, pp. 322-373, Sep. 2001.

[7] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," In *Proc Int'l. Conf. on Very Large Data Bases, VLDB*, pp. 426-435, 1997.

[8] Ö. Egecioglu, "Parametric Approximation Algorithms for High-Dimensional Euclidean Similarity," In *Proc. European Conf. on*

Principles of Data Mining and Knowledge Discovery, PKDD, pp. 79-90, Sep. 2001.

[9] Ö. Egecioglu, H. Ferhatosmanoglu, and U. Ogras, "Dimensionality Reduction and Similarity Computation by Inner Product Approximations," In *IEEE Trans. on Knowledge and Data Engineering*, pp. 714-726, 2004.

[10] H. Eidenberger, "A New Method for Visual Descriptor Evaluation," In *Proc. SPIE Storage and Retrieval Methods and Applications for Multimedia*, pp. 145-157, 2004.

[11] C. Faloutsos, R. Barber, M. Flickner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and Effective Querying By Image Content," In *Journal of Intelligent Information Systems*, Vol. 3 No. 3/4 pp. 231-262, Jul. 1994.

[12] S. Jeong, S. Kim, K. Kim, and B.-U. Choi, "An Effective Method for Approximating the Euclidean Distance in High-Dimensional Space," In *Journal of the Institude of Electronics Engineers of Korea*, Vol. 42-CI No. 5 pp. 69-78, 2005.

[13] K. V. R. Kanth, D. Agrawal, and A. Singh, "Dimensionality Reduction for Similarity Searching in Dynamic Databases," In *Proc. Int'l. Conf. on Management of Data, ACM SIGMOD*, pp. 166-176, Jun. 1998.

[14] N. Katayama and S. Satoh, "The SR-Tree: An Index Structure for High-dimensional Nearest Neighbor Queries," In *Proc. Int'l. Conf. on Management of Data, ACM SIGMOD*, pp. 369-380, 1997.

[15] S. Krishnamachari and M. Abdel-Mottaleb, "Hierarchical Clustering Algorithm for Fast Image Retrieval," In *Proc. IS & T/SPIE Conf. on Storage and Retrieval for Image and Video Databases*, pp. 427-435, Jan. 1999.

[16] K. Lin, H. Jagadish, and C. Faloutsos, "The TV-Tree: An Index Structure for High Dimensional Data," *The VLDB Journal*, Vol. 3, No. 4, pp. 517-542, 1994.

[17] A. Mertins, *Signal Analysis*, John Wiley & Sons, Inc., 2000.

[18] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice-Hall, 2000.

[19] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker, "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape," In *Proc. Int'l. Conf. Storage and Retrieval for Image and Video Databases*, pp. 173-187, 1993.

[20] U. Ogras and H. Ferhatosmanoglu, "Dimensionality Reduction Using Magnitude and Shape Approximations," In *Proc. Int'l. Conf. on Information and Knowledge Management, ACM CIKM*, pp. 99-107, 2003.

[21] B.-U. Pagel, H-W. Six, and M. Winter, "Window Query-Optimal Clustering of Spatial Objects," In *Proc. Int'l. Conf. on Principals of Database Systems*, pp. 86-94, 1995.

[22] T. Seidl and H.-P. Kriegel, "Efficient User-daptable Similarity Search in Large Multimedia Databases, In *Proc. Int'l. Conf. on Very Large Data Bases, VLDB*, pp. 506-515, Aug. 1997.

[23] T. Seidl and H.-P. Kriegel, "Optimal Multi-Step k-Nearest Neighbor Search," In *Proc. Int'l. Conf. on Management of data, ACM SIGMOD*, pp. 154-165, June 1998.

[24] R. Weber, H. J. Schek, and S. Blott, "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces," In *Proc. Int'l. Conf. on Very Large Data Bases, VLDB*, pp. 194-205, Aug. 1998.

[25] D. A. White and R. Jain, "Similarity Indexing with the SS-tree," In *Proc. Int'l. Conf. on Data Engineering, IEEE*, pp. 516-523, 1996.

[26] http://kdd.ics.uci.edu/databases/CorelFeatures/CorelFeatures.html

저 자 소 개

정 승 도(학생회원)

1999년 한양대학교 전자·전자통신·전파공학과 학사 졸업
2001년 한양대학교 전자통신전파공학과 석사 졸업
2001년 ~ 현재 한양대학교 전자통신컴퓨터공학과 박사과정 재학 중

<주관심분야 : 컴퓨터비젼, 생체인식, IBR , AR>

김 상 욱(정회원)

1989년 서울대학교 컴퓨터공학과 학사 졸업
1991년 한국과학기술원 전산학과 석사 졸업
1994년 한국과학기술원 전산학과 박사 졸업

현 한양대학교 정보통신대학 정보통신학부 교수

<주관심분야 : 데이타베이스 시스템, 저장 시스템, 데이타 마이닝, 멀티미디어 정보 검색, 공간 데이터베이스/GIS, 주기억장치 데이타베이스, 트랜잭션 관리>

최 병 욱(정회원)

1973년 한양대학교 전자공학과 학사 졸업
1978년 일본 경응의숙(KEIO)대학 전기공학과 석사 졸업
1981년 일본 경응의숙(KEIO)대학 전기공학과 박사 졸업

현 한양대학교 정보통신대학 정보통신학부 교수

<주관심분야 : 영상처리, 멀티미디어 공학>