Journal of Korea Multimedia Society Vol. 15, No. 2, February 2012(pp. 215-225)

영상 색인용 VP-tree의 검색 범위 압축법의 개선에 관한 연구

박길양†, 이상곤††, 황재정†††

요 약

멀티미디어 데이터베이스에서는 검색 효율을 높이기 위해 다차원 공간에 기초한 색인 방법이 사용되고 있다. 그러나 이 방법은 거리 계산의 척도로 유클리드 거리를 이용하여야 한다는 전제가 있어 범용성이 떨어진다. 한편, 거리 공리의 성립을 전제로 하는 거리 공간에 기반한 색인 방법은 유클리드 거리 이외의 거리 척도를 이용할 수 있기 때문에 범용성이 높다. 본 논문에서는 거리 공간을 색인화하는 방법 중 하나인 VP-tree의 방법을 개선하고자 한다. VP-tree는 검색 시에 루트 노드로부터 검색 범위에 적합한 노드를 따라 최종에 이르는 리프 노드에 링크되어 있는 오브젝트와의 거리를 계산하고, 검색 범위에 적합한가를 검사한다. 그러나 리프 노드에서 거리 계산 횟수가 증가하면 검색 속도가 떨어지기 때문에 리프 노드에서 삼각 부등식을 이용한 범위 압축 방법에 주목하고 그 개량 방법으로서 질의 오브젝트에 대한 최근접점을 삼각 부등식의 기준점으로 이용하는 방법을 제안한다. 이 개량 방법에 의해 검색 범위를 크게 좁힐 수 있으며, 또한 거리 계산의 횟수도 꽤 줄일 수 있다. 실제로 10,000 건의 영상 데이터를 이용하여 시스템의 성능 평가를 진행해 본 결과 기존 방법에 비해 유사 영상의 검색 시간을 5%~12%까지 절감할 수 있었다.

Study of Improvement of Search Range Compression Method of VP-tree for Video Indexes

Gil-Yang Park†, Samuel Sangkon Lee††, Jea-Jeong Hwang†††

ABSTRACT

In multimedia database, a multidimensional space-based indexing has been used to increase search efficiency. However, this method is inefficient in terms of ubiquity because it uses Euclidean distance as a scale of distance calculation. On the contrary, a metric space-based indexing method, in which metric axiom is prerequisite is widely available because a metric scale other than Euclidean distance could be used. This paper is attempted to propose a way of improving VP-tree, one of the metric space indexing methods. The VP-tree calculates the distance with an object which is ultimately linked to the a leaf node depending on the node fit for the search range from a root node and examines if it is appropriate with the search range. Because search speed decreases as the number of distance calculations at the leaf node increases, however, this paper has proposed a method which uses the latest interface on query object as the base point of trigonometric inequality for improvement after focusing on the trigonometric inequality-based range compression method in a leaf node. This improvement method would be able to narrow the search range and reduce the number of distance calculations. According to a system performance test using 10,000 video data, the new method reduced search time for similar videos by 5-12%, compared to a conventional method.

Key words: Multi-dimensional DB(다차원 DB), Multimedia DB(멀티미디어 DB), Main Storage DB(주기억 DB), VP-Tree(VP-트리)

※ 교신저자(Corresponding Author) : 이상곤, 주소 : 전라북도 전주시 완산구 천잠로 303번지(560-759), 전화 : 063) 220-2934, FAX : 063) 220-2056, E-mail : samuel@jj.ac.kr
접수일 : 2011년 08월 01일, 수정일 : 2011년 12월 22일
완료일 : 2012년 02월 01일

† 정회원, 군산대학교 전자정보공학부
(E-mail : gy7500@gmail.com)
†† 종신회원, 전주대학교 컴퓨터공학과
††† 정회원, 군산대학교 전파공학과
(E-mail : hwang@kunsan.ac.kr)

1. 서 론

근래에는 일차 혹은 이차 기억장치의 가격이 낮고, 대용량화에 의해 개인용 컴퓨터에도 문서, 화상, 음악, 영상 등 멀티미디어 데이터의 대량 보존이 가능해졌다. 그에 따라 대량으로 보존된 멀티미디어 데이터에서 사용자가 원하는 데이터만을 신속하고 정확하게 검색하는 기술이 필요해졌다. 검색 효율을 높이기 위해서는 사전에 저장된 데이터로부터 특징을 추출하고, 그 특징으로부터 색인(인덱스)을 구성할 필요가 있다[1]. 실제 검색 시에는 그 인덱스에서 적합한 데이터를 추출한다. 그렇기 때문에 인덱스의 구성 방법에 따라 검색 시스템의 효율이 크게 좌우된다.

멀티미디어 데이터에서 추출한 특징은 일반적으로 벡터(vector)로 표현되는데, 각각의 특징 벡터 사이의 거리가 유사한 정도에 따라 검색 결과가 달라진다[2,3]. 이와 같이 특징 벡터의 색인화 방법에 따라, 예컨대 다차원 데이터의 인덱스 구성 방법에 대한 연구는 R-tree[4], R*-tree[5], SS-tree[6], SR-tree[7], X-tree[8], VA-FILE[9] 등이 제안되어 있다. 그러나 이와 같은 방법은 유클리드 거리를 척도로 이용하여야 한다는 전제가 있어 그 밖의 거리 계산 방법[10]에는 적용이 불가능하다. 여기서 제기한 유클리드 거리 이외에 사용 가능한 거리 척도 방법으로 다차원 데이터의 각 차원 간의 상관관계를 고려한 Quadratic Form 거리[11], 문자열 사이의 유사성을 계산하는 Edit 거리, 영상 간의 구도 유사성을 계산하는 Earth Mover's Distance[12] 등이 제안되어 있으나 여전히 여러 문제점이 존재한다.

거리 정보에 기초하여 색인 작업을 수행하는 거리 공간 인덱스에 대한 연구가 중요하다. 다차원 인덱스가 다차원 공간상의 특징 좌표값을 기반으로 인덱스를 작성한다는 점에 비해, 거리 공간 인덱스는 거리 공리(距離 公理)가 성립하면 특징 사이의 거리 정보만을 이용하여 비교적 간단하게 색인 작업이 가능하다. 따라서 유클리드 거리 이외의 거리 계산 방법에도 적용 가능하다. 거리 공간 인덱스는 일반적으로 계층적 인덱스 트리이며, 공간(데이터 집합)을 거리 정보에 기반하여 재귀적으로 분할함으로서 탐색 시의 검색 공간을 축소할 수 있다. 이 공간 분할법의 차이에 따라 M-tree[13], VP-tree[14,15], MVP-tree[16], MI-tree[17] 등이 제안되어 있다. M-tree는 공간 분할 시에 상향적(Bottom-up)인 인덱스 트리를 구성하기 때문에 분할한 공간 사이에 공통 영역이 많아 검색 효율이 저하되는 단점이 있다. 이에 반해, VP-tree는 Vantage Point 라 불리는 기준점을 이용하여 탐색 공간을 하향적(top-down)으로 분할하기 때문에 분할 공간 내에 공통 영역을 생성하지 않아도 된다. 검색 시에는 루트 노드로부터 검색 범위에 적합한 노드를 따라 최종에 이르는 리프 노드에 링크된 리프 오브젝트에 차례로 접근하여 거리를 계산하고, 검색 범위에 적합한가를 조사한다. 그러나 검색 시에 순회한 리프 노드의 거리 계산 횟수를 증가시켜 검색 속도를 떨어뜨리는 원인이 된다.

본 논문에서는 VP-tree의 리프 노드에서 검색 알고리즘을 개선하여 거리 계산 횟수의 절감을 증명하고자 한다. VP-tree의 리프 노드에서 삼각 부등식[1]을 이용한 검색 범위의 압축에 대한 종래의 방법은 Vantage Point를 삼각 부등식의 기준점으로 이용하는 것에 반해, 본 논문의 방법은 삼각 부등식의 기준점과 질의 오브젝트간의 거리가 가까우면 가까울수록 해석되는 범위가 줄어드는 점에 주목하였다. 따라서 본 방법에서는 삼각 부등식의 기준점에 질의 오브젝트에 대한 최근접점을 이용함으로서 검색 범위를 현저하게 좁히고, 거리 계산의 횟수를 대폭 줄일 수 있다. 다만, 본 논문에서 제안하는 개량 방법을 실제 시스템에 도입하여도 최근접점을 사전에 적용할 수는 없기 때문에, 본 방법에서는 검색 결과 리스트 내에 질의 오브젝트에 가장 가까운 오브젝트를 가상의 최근접점(virtual nearest point)으로 가정하여 최근접점을 이용한 범위에 대한 압축 기술을 실현하였다. 또한 가상의 최근접점을 기준점으로 하여 삼각 부등식을 이용하기 위해서는 가상의 최근접점과 리프 노드 내의 모든 오브젝트 사이의 거리를 미리 알고 있어야 한다. 여기서 가상의 최근접점을 사전에 결정할 수는 없기 때문에 실제적으로 모든 오브젝트 간의 거리 계산이 필요하다. 따라서 본 논문에서는 인덱싱

1) 삼각 부등식(三角 不等式, trigonometric inequality)은 삼각형의 세 변에 대한 부등식으로, 임의의 삼각형의 두 변의 길이의 합은 나머지 한 변의 길이보다 크다는 것이다. 이 부등식은 여러 공간에 적용된다. 먼저 노름 벡터 공간에서는 두 변의 벡터를 각각 x, y 라고 하면 이 부등식은 다음과 같이 쓸 수 있다. $\| x + y \| \leq \| x \| + \| y \|$. 또한, 거리 공간 M 에 x, y, z 가 있고, 이들 사이의 거리를 d 라고 한다면 다음의 부등식이 성립한다. $d(x, z) \leq d(x, y) + d(y, z)$

시에 오브젝트 간의 거리를 계산한 거리 리스트 파일(거리 목록 파일, distance list file)을 구축한다. 대규모의 거리 목록 파일을 오브젝트마다 분할하여 구축함으로서 메모리로 읽어 들이는 거리 목록 파일의 크기도 줄일 수 있다.

본 논문에서 연구하고자 하는 방법과 유사한 연구로 사전에 구축한 거리 리스트 파일을 이용하여 해석될 범위의 압축을 실행하는 AESA; Approximating and Eliminating Search Algorithm[18]가 있다. 그러나 AESA와 본 논문에서 제안하는 방법과의 차이점은 AESA가 모든 오브젝트를 대상으로 거리 리스트를 이용한 검색 범위의 압축을 시도하는 것에 비해, 본 논문의 방법은 리프 노드 내의 오브젝트만을 대상으로 한다. 모든 오브젝트를 대상으로 하는 AESA는 필연적으로 파일을 불러드리는 횟수가 급격하게 증가하는 문제가 발생한다. 반면 본 논문의 방법과 같이 VP-tree를 이용하면 리프 노드 내의 몇 개의 오브젝트만이 대상이 되며, 가상의 최근접점(기준점)이 갱신되는 경우에만 거리 목록 파일을 불러오기 때문에 파일의 접속 횟수를 크게 줄일 수 있다.

※ VP-tree의 구축 알고리즘
(1) 데이터 집합으로부터 무작위로 가상의 vp을 선택한다.
(2) 가상 vp 부터 나머지 $N-1$ 개의 오브젝트까지의 거리를 계산한다.
(3) 이들 사이의 거리의 중간값(mean)과 분산(variance)을 계산한다.
(4) (1)~(3)을 몇 차례 반복하여 분산이 최대에 이르는 지점을 vp로 결정한다.

이하 2장에서는 VP-tree의 구축과 검색 알고리즘을 설명한 후, 리프 노드의 범위 압축 방법에 대해 설명한다. 그리고 제3장에서 리프 노드의 검색 알고리즘의 개량법을 소개한다. 제4장에서는 이 개량법을 이용한 실험과 평가에 대해 서술한다. 마지막으로 5장에서는 결론을 서술한 뒤 향후 과제와 연구의 전망에 대해 기술한다.

2. 개선된 VP-tree

2.1 구축 알고리즘

이 장에서는 VP-tree의 구축 알고리즘에 대해 설명한다. N개의 데이터로 구성된 데이터의 집합 S에 대한 색인 작업을 수행한다고 가정하였을 때, 트리의 각 노드는 아래에 나타난 바와 같은 무작위 알고리즘에 따라 vantage point[2](이하, 'vp'라 기술)를 선정할 수 있다.

루트 노드에 선택된 vp로부터 S 사이의 모든 데이터의 거리에 대한 중간값을 μ라 한다. $d(p, q)$를 점 p, q 사이의 거리(distance)라 하면, 데이터 집합 S는 아래와 같이 두 개의 영역 S_1과 S_2로 분할된다.

$$S_1 = \{s \in S \mid d(s, vp) < \mu\},$$
$$S_2 = \{s \in S \mid d(s, vp) \geq \mu\}$$

이와 같은 분할 작업을 S_1 및 S_2의 영역에 재귀적으로 적용함으로서 인덱스를 생성한다. S_1과 S_2의 모든 부분 집합은 VP-tree의 한 개의 노드에 상응한다. 리프 노드에는 몇 개의 오브젝트들을 저장하고 있다.

2.2 검색 알고리즘

본 절에서는 VP-tree의 범위 지정 검색(range search) 및 K-최근린 검색(K-최근점 검색, K-nearest neighbor search)의 알고리즘에 대해 설명한다. 범위 지정 검색이란 질의 오브젝트와 검색 범위 사이를 원의 반경을 이용하여 원의 중심으로부터 반경까지의 거리에 존재하는 오브젝트의 집합을 구하는 검색 방법이다. 또한 K-최근린 검색이란 질의 오브젝트 및 검색 건수 K를 지정하고 거리가 가까운 순으로 상위 K건의 오브젝트의 집합을 구하는 검색 방법이다. 본 논문에서는 K-최근린 검색을 이용한 실험을 진행하였으나, 이 검색 방법이 범위 지정 검색 알고리즘에 기초하고 있기 때문에 본 절에서는 이들 두 가지 검색 방법 모두에 대해 서술하고자 한다.

범위 지정 검색은 루트 노드로부터 검색 범위에 적합한 노드를 따라 리프에 링크되어 있는 리프 오브젝트와 질의 오브젝트와의 거리를 계산하고, 검색 범위에 존재하는 오브젝트를 얻는다. 한편, K-최근린 검색은 검색 초기 값으로 검색 반경을 무한대로 설정하여 루트를 따라 오브젝트를 검색 결과 리스트에 계속하여 추가해 나간다. 검색 결과 리스트의 검색 수가 지정된 검색 수를 넘기면 거리가 최대인 검색

2) 사전적인 의미는 다음과 같다. (무엇을 지켜보기에) 좋은 위치; (특히 과거를 생각해 보는) 시점.

오브젝트를 검색 결과 목록에서 삭제하고, 검색 결과 리스트의 검색 수가 지정된 검색 수를 넘지 않도록 한다. 또한 검색 결과 리스트의 최대 거리를 검색 반경으로 하여 반복 수행함으로서 검색 반경을 좁혀 검색하고, 최종적으로 지정된 건수만큼의 분량을 검색 결과로 얻을 수 있다.

2.3 리프 노드에서 검색 범위의 압축

2.2절에 서술한 바와 같이, 기존의 VP-tree에서는 검색 중에 리프 노드 내에 존재하는 모든 오브젝트에 접근하여 질의 오브젝트와의 거리를 계산하였다. 그러나 이를 개선하는 또 다른 방법으로 리프 노드 내의 각 오브젝트를 검색할 때에 삼각 부등식을 이용하면 다음과 같이 검색 후보의 범위를 축소하는 방법[19]이 기존 연구로 제안되어 있다.

리프 노드에서 리프 노드의 vp 오브젝트와 각 리프 오브젝트 사이의 거리를 거리 리스트(distance list)로 보존한다. 이 vp 오브젝트와 각 리프 오브젝트 사이의 거리를 삼각 부등식을 이용하면 거리 계산의 횟수를 줄일 수 있다. 질의 오브젝트를 q, 검색 범위에 있는 반경을 r, 리프 노드의 vp 오브젝트를 v, 리프 노드에 링크하는 리프 오브젝트를 o라 하면, 아래와 같은 定理가 성립한다.

定理(Theorem) ① : $d(v, o) - d(v, q) > r$ 가 성립한다면, 리프 오브젝트 o는 검색 범위 내에 존재하지 않는다.
[증명] 삼각 부등식 $d(v, q) + d(q, o) \geq d(v, o)$ 에 의해 $d(v, o) - d(v, q) > r$ 은 $d(q, o) > r$ 이 되며, o는 검색 범위에 존재하지 않음을 알 수 있다. $-d(v, o) + d(v, q) > r$ 도 마찬가지로 $d(q, o) > r$ 이 된다. 따라서 위의 정리 ①은 성립한다.

定理 ①의 $d(v, o)$ 및 r은 리프 노드의 검색 시에 이미 알고 있는 정보이며, $d(v, q)$ 는 각 리프 노드를 한 번만 구할 수 있기 때문에 각 리프 오브젝트와의 거리를 일일이 계산하지 않아도 리프 오브젝트가 검색 범위에 존재하는지 혹은 존재하지 않는지 판단할 수 있다. 따라서 거리 계산 횟수와 리프 오브젝트의 접근 횟수를 줄일 수 있다. 이 리프 노드의 후보(해석 후보)를 좁힐 수 있는 방법을 다음의 그림 1에 나타냈으며, K-최근린 검색 방법의 알고리즘을 다음과 같이 제시하였다. 그림 1에서 사선 이외의 부분은 定理 ①의 식이 성립하는 부분이며, 이 부분에 존재하는 오브젝트들의 거리 계산은 생략 가능하다. 한편, 사선 부분은 정리 ①이 성립하지 않는 부분이며, 이 부분에 존재하는 오브젝트들은 모두 거리 계산이 필요하다.

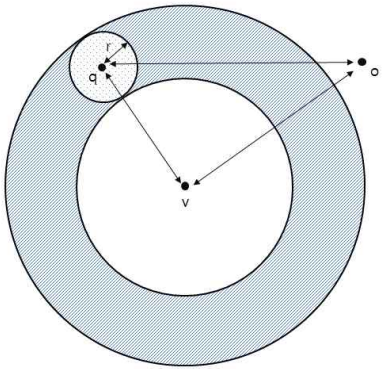

그림 1. *vp*를 기준점으로 한 검색 범위의 압축

※ K-최근린 검색 방법을 이용한 리프 노드의 검색 알고리즘

```
입력 : q, r, L,
출력 : L,
Search_Leaf (q, r, L)
 {
    foreach o (모든 리프 오브젝트) {
      if ( |d(v, o) − d(v, q)| ≤ r ) {
          if ( d(o, q) ≤ r ) {
                L에 o를 더해
                 거리의 최대값을 r
                 이라 한다.
                }
          }
    }
 }
```

여기서, q : 질의 오브젝트, r : 검색 반경, o : 리프 오브젝트, v : vp 오브젝트, L : 검색된 결과 리스트이다.

또한 리프 노드의 vp 오브젝트뿐만 아니라, 루트 노드에서 리프 노드까지에 이르는 패스 상에 존재하는 모든 vp 오브젝트를 이용하여도 검색 범위의 압축이 가능하다. 이 경우 리프 노드에 링크된 모든 리프 오브젝트와 루트 노드에서 그 리프 오브젝트까지 패

스 상에 존재하는 모든 vp 오브젝트까지의 거리를 리프 노드보다 앞서 저장할 필요가 있다. 이 복수의 vp 오브젝트와 각 리프 오브젝트 사이의 거리를 삼각 부등식을 이용하여, 거리 계산 횟수를 줄일 수 있다. 질의 오브젝트를 q, 검색 범위인 반경을 r, 루트 노드부터 리프 노드까지의 패스 상에 존재하는 k개의 vp 오브젝트를 $v_i\,(i=1, 2, 3, \ldots, k)$, 리프 노드에 링크하는 리프 오브젝트를 o 라고 하면, 그림 2와 같이 해석 후보를 줄일 수 있다. 이 알고리즘은 여러 개의 vp 오브젝트를 이용하여 비교하므로 해석 후보를 기존의 방법보다 쉽게 좁힐 수 있을 가능성이 높아진다.

定理 ② : $d(o_1, o) - d(o_1, q) > r$ 가 성립한다면, 리프 오브젝트 o은 검색 범위에 존재하지 않는다.
[증명] 삼각 부등식 $d(o_1, q) + d(q, o) \geq d(o_1, o)$ 에 의해 $d(o_1, o) - d(o_1, q) > r$ 은 $d(q, o) > r$ 이 되며, o가 검색 범위 안에 존재하지 않음을 알 수 있다.

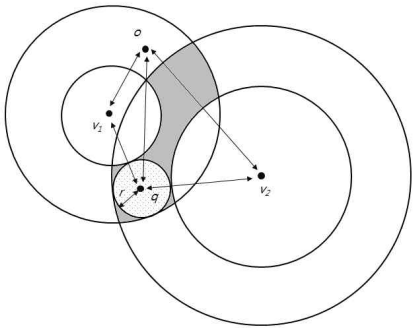

그림 2. 여러 개의 *vp*를 기준점으로 하는 경우에서 검색 범위의 압축

3. 최근접점을 이용한 검색 범위의 압축 방법

지금까지 설명한 vp를 삼각 부등식의 기준점으로 이용하는 경우, 정리 ①이 성립하지 않는 부분(그림 1의 사선 부분)이 적을수록 검색 범위의 압축 효과가 향상된다. 이 부분의 외주원의 반경은 vp로부터 질의 오브젝트인 q까지의 거리에 검색 범위인 반경 r을 더한 값이 된다. 여기서 r은 검색 요구에 따른 고정값이기 때문에 vp와 q와의 거리가 좁을수록 검색 범위의 압축이 유리하게 된다. 덧붙여 말하면 q에 가장 근접한 오브젝트가 최근접점이다. 즉, vp 대신에 q의 최근접점인 오브젝트를 삼각 부등식의 기준점으로 이용함으로써 정리 ①이 성립하지 않는 부분을 검색 대상 영역에서 제거하여 검색 영역을 좁힐 수 있다. 따라서 본 논문에서는 최근접점을 기준점으로 하는 삼각부등식을 이용한 검색 범위의 새로운 압축 방법을 제안하고자 한다.

질의 오브젝트를 q, 검색 범위인 원의 반경을 r, 검색 리스트 내의 질의 오브젝트에 가장 가까운 최근접점을 o_1, 리프 노드에 링크된 리프 오브젝트를 o라 하면, 定理 ②가 성립한다.

여기서 $d(o_1, o)$, $d(o_1, q)$를 미리 알고 있다면, 각 리프 오브젝트와의 거리를 일일이 계산하지 않고도 오브젝트가 검색 범위 내에 존재하지 않는다는 사실을 알 수 있다. 이 과정을 그림 3에 나타내었다. 그림에서와 같이 사선 부분에 리프 오브젝트(그림에서 o가 사선 부분 바깥에 있음)가 존재하지 않으면 질의 오브젝트와의 거리 계산을 생략할 수 있다. 실제로 리프 노드의 검색 알고리즘을 다음에 제시하였다.

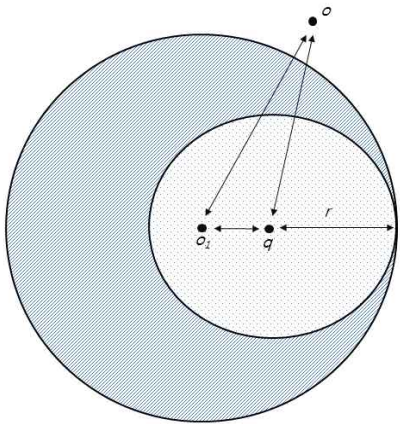

그림 3. 최근접점을 기준점으로 하는 검색 범위의 압축

※ 본 논문의 방법을 이용한 리프 노드의 검색 알고리즘

```
입력 : q, r, L
출력 : L
Search_Leaf(q, r, L)
    {
        foreach o (모든 리프 오브젝트) {
            if ( (d(o1, q) + r) < d(o1, o) ) {
```

```
            if ( d(o, q) ≤ r ) {
                    L에 o를 더해
                     거리의 최대값을 r
                     이라 한다.
                    }
                }
        }
    }
```

여기서, q : 질의 오브젝트, r : 검색 반경, o : 리프 오브젝트, o_1 : 검색 리스트 안에서 q에 가장 가까운 오브젝트, L : 검색 결과 리스트 등이다.

> 定理 ③ :
> $d(s, o) - d(s, q) > r$ 가 성립한다면, 리프 오브젝트 p는 검색 범위에 존재하지 않는다.

정리 ②의 $d(o_1, o)$는 최근접점과 리프 노드 내의 오브젝트의 거리 목록에 존재하면 값이 부여된다. 단, 어떤 오브젝트가 q의 최근접점인지 사전에 알 수 없기 때문에 실제로 모든 오브젝트가 o_1의 후보어가 될 수 있다. 따라서 색인을 위해 리프 노드에 링크된 리프 오브젝트와 그 밖의 다른 오브젝트와의 거리를 계산한 거리 목록 파일을 작성할 필요가 있다. 그러나 이와 같은 방대한 색인 파일을 메모리상에 구축하는 것은 좀 곤란하다. 따라서 본 방법에서는 Akama의 방법[20]과 같이, 각 오브젝트 ID를 파일 이름으로 하는 그림 4와 같이 파일 집합을 구축하였다. 이 그림은 거리 목록 파일이 보조기억장치에 표현된 것을 보여주는 것이다. 컴퓨터 운영체제의 한 디렉터리 내의 최대 파일수의 제한을 피하기 위해 ID의 아래부터

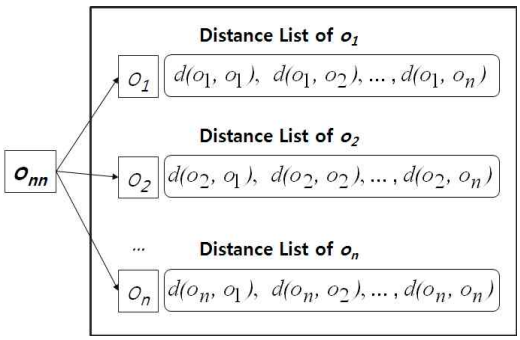

그림 4. 거리 목록 파일의 구축 예

세 자릿수씩 구분하여 최하위의 것을 파일 이름, 상위의 것을 디렉터리 이름으로 하였다. 또한 적절한 입출력 속도를 얻기 위해 한 개의 디렉터리 내의 최대 파일의 수를 1,000개 이하로 하였다.

또한, 질의 오브젝트 q와 최근접점 o_1과의 거리인 $d(o_1, q)$도 마찬가지로 최근접점 자체를 사전에 결정할 수는 없기 때문에, 본 논문의 방법에서는 위의 리프 노드에 의한 검색 알고리즘에서 제시한 바와 같이, 검색 결과 리스트 L 안에서 질의 오브젝트 q에 가장 가까운 오브젝트를 최근접점 o_1이라 가정하였다. 그리고 검색 범위 안에 존재하는 오브젝트를 새롭게 발견할 때 마다 최근접점 o_1을 다시 갱신하는 전략을 채택하였다.

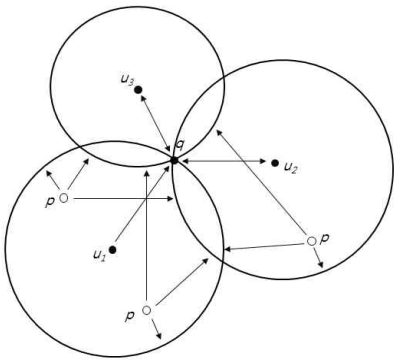

그림 5. 기준점 s 를 선택하는 방법

4. 실험 및 평가

4.1 실험 방법

본 논문의 개선 방법을 VP-tree에 실제로 적용하여 이를 이용한 유사 영상 검색에 적용하는 실험을 하였다. 실험에서 OS는 Linux, CPU는 Pentium D 3.2 GHz, 메모리는 2 GB 사양의 컴퓨터를 이용하였다. 우선 등록할 영상으로 10,000 건의 영상을 준비하고, 영상의 특징으로 HSI 히스토그램을 이용하여 특징을 추출한다. HSI 히스토그램은 색상(hue), 채도(saturation), 광도(intensity)로 구성된 색 히스토그램이다. 히스토그램의 차원 수는 12(4×3) 차원, 24(8×3) 차원, 48(16×3) 차원, 96(32×3) 차원 등 총 네 종류의 특징을 이용하였다. 덧붙여 특징 추출을 끝낸 영상 오브젝트 10,000 건을 앞에서 언급한 VP-tree로 색인하였다. 인덱싱 시의 vp은 매 회 최대 100

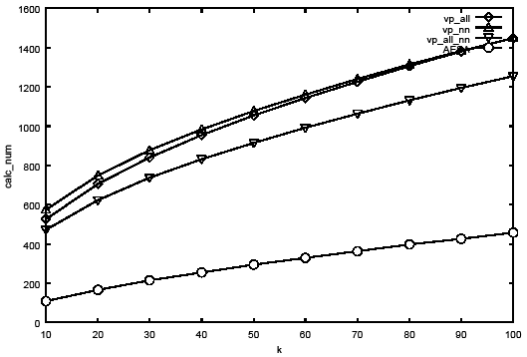

그림 6. 12차원 데이터에 대한 거리 계산 회수

건의 랜덤 데이터를 기준으로 계산하였다. 인덱싱에 이용하지 않은 1,000 개 가량의 입력 영상에서 K-최근린 검색 방법을 이용하여 검색 시에 요구되는 거리 계산 횟수 및 CPU 시간의 영상 1건 당 평균값을 구하였다.

또한 영상 오브젝트 사이의 거리 척도로 quadratic form 거리를 이용하였다. 히스토그램 H와 히스토그램 K 사이의 quadratic form 거리는,

$$D_q(H,\ K) = \sqrt{(h-k)^T A(h-k)} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}(h_i-k_i)\,(h_j-k_j)} \quad (1)$$

로 표시된다. 여기서 행렬 $A=[a_{ij}]$ 는 히스토그램의 i 번째 변과 j 번째 변의 유사도를 나타내는 행렬(matrix)이다. 본 논문에서는 아래의 식 (2)와 같은 행렬식[19]을 이용했다.

$$a_{ij} = 1 - \frac{d(i,j)}{d_{\max}} \quad (2)$$

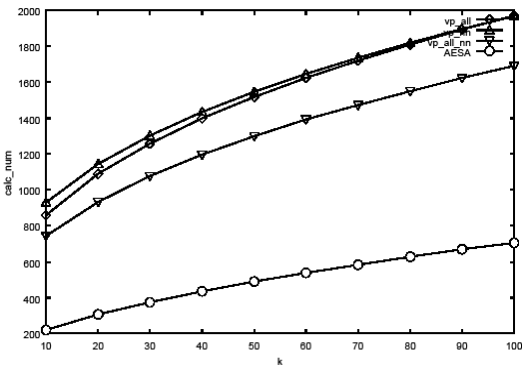

그림 7. 24차원 데이터에 대한 거리 계산 회수

위의 식 (2)에서 $d(i,j)$ 는 i 번째와 j 번째 변의 색 공간상의 거리(색 차이)이며, $d_{\max}$는 $d(i,j)$ 의 최대값이다. 덧붙여 본고에서는 VP-tree의 최근접점을 이용한 범위의 압축 방법을 제안하고 있으나, 최근접점을 이용하여 검색 범위를 압축하는 알고리즘으로 앞의 1장 끝부분에서 언급한 AESA[18]도 존재한다. AESA는 VP-tree와 같은 인덱스 트리를 구축하지 않고, 각 오브젝트 사이의 거리를 계산한 파일을 미리 작성하고 이를 이용하여 검색 대상이 되는 검색 범위를 결정한다. 또한 검색 범위의 압축에는 본 논문에서 제안하는 방법과 동일하게 최근접점을 삼각 부등식을 이용하여 압축한다. 본 논문이 제안하는 방법과 AESA는 대단히 유사한 알고리즘이지만 그 차이점을 설명하면 다음과 같다. 본 논문에서는 VP-tree에서 최근접점을 이용한 검색 범위의 압축 개량법과 비교 대상으로서 이 AESA를 이용하여 검색 범위의 압축 효과에 관한 우열을 평가한다. 이를 설명하기 위해 AESA의 구체적인 알고리즘을 다음 절에서 소개한다.

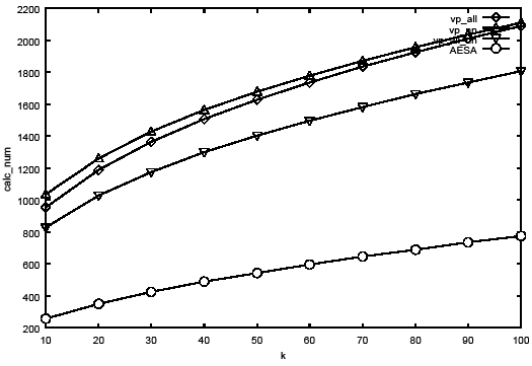

그림 8. 48차원 데이터에 대한 거리 계산 회수

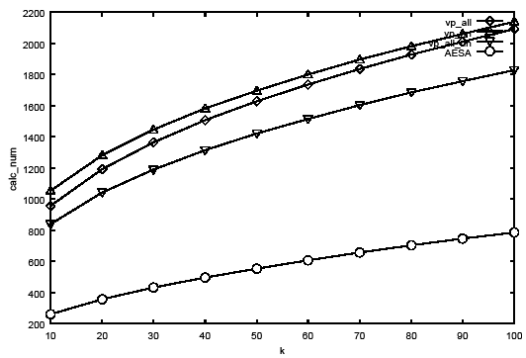

그림 9. 96차원 데이터에 대한 거리 계산 회수

4.2 AESA

AESA는 최근접점 s를 아래의 식 (3)에 의해 예측한다.

$$s = \arg\min_{\forall o \in P-E} \sum_{\forall o \in U} |d(o,u) - d(q,u)| \qquad (3)$$

P는 모든 오브젝트의 집합, E는 검색 대상 이외로 판정되어 제거된 오브젝트 집합, U는 과거에 최근접점 s로 선정된 오브젝트 집합이며, q는 질의 오브젝트이다. 이 식의 내용을 그림 5에 설명하였다.

각 u를 중심으로 $d(q,u)$을 반지름으로 하는 원 C1. C2, C3(그림에서 u_1을 중심으로 하는 원을 C1, 그 오른쪽 원이 C2, 가장 위의 원을 C3)라 가정했을 때, $\sum |d(o,u) - d(q,u)|$는 각 원과 o와의 거리의 합계가 된다. 이 거리의 합계가 가장 작아지는 o가 s로 선택된다. 즉, 각 u에서 보았을 때 q에 가장 가깝다고 예상되는 o를 산출한다. 이로서 질의 오브젝트와 전체 오브젝트간의 거리를 직접 계산하지 않아도 그 시점에서 q에 가장 가깝다고 생각되는 점의 계산이 가능하다.

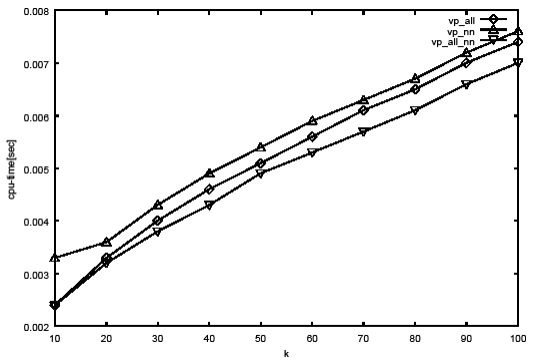

그림 10. 12차원 데이터에 대한 실행 시간

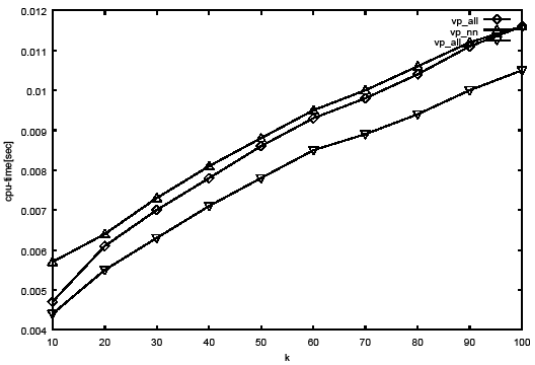

그림 11. 24차원 데이터에 대한 실행 시간

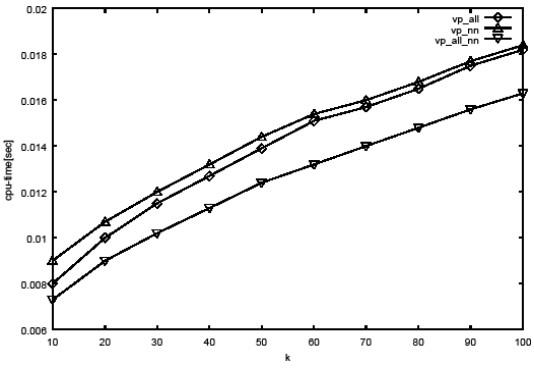

그림 12. 48차원 데이터에 대한 실행 시간

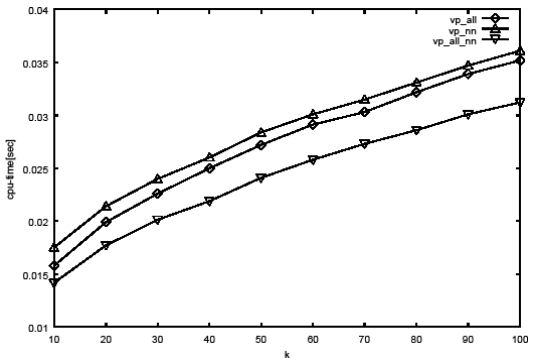

그림 13, 96차원 데이터에 대한 실행 시간

위와 같은 방법으로 선택된 s와 질의 오브젝트 q와의 거리를 계산하고 이 거리가 검색 결과 리스트에 포함될 것 같으면 검색 결과 리스트에 삽입한다. 또한 과거 최근접점(검색 결과 리스트 내의 1위 후보)과 q와의 거리보다 가깝다면 이 s을 최근접점으로 새롭게 갱신한다. 마지막으로 삼각 부등식에 의해 검

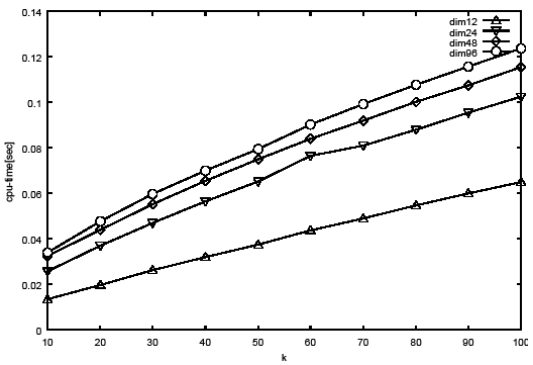

그림 14. AESA의 실행 시간

색 범위의 압축 작업을 계속한다. 정리 ①과 정리 ②와 마찬가지로 정리 ③도 성립한다.

여기서, 건수 지정 검색의 검색 반경 r은 질의 오브젝트 q와 검색 결과 리스트 최하위에 위치한 오브젝트와의 거리이다. 또한 $d(s, o)$는 사전에 작성된 각 오브젝트 간의 거리 리스트 파일에서 읽어 갱신할 수 있다. $d(s, q)$는 직전 최근접점의 갱신 처리에서 계산된다. 따라서 정리 ③에서 사용되는 모든 거리 정보는 삼각 부등식 적용 시에 미리 알 수 있는 정보이며, 각 오브젝트와 q와의 거리를 일일이 계산하지 않고도 오브젝트가 검색 범위 안에 존재하지 않는다고 판단할 수 있다. 이상과 같이 본 논문의 방법을 이용하면 불필요한 오브젝트와의 거리 계산 과정을 생략할 수 있으며, 거리 계산 횟수 또한 줄일 수 있다. 이상의 처리 과정을 모든 오브젝트가 제거될 때까지 반복함으로서 최종적으로 우리가 원하는 빠른 검색 결과를 얻을 수 있다.

4.3 실험 결과

각 검색 범위의 압축을 위한 개선 방법들을 이용하여 건수 지정 검색 방법에 대한 실험을 다음과 같이 진행하였다. 그래프 안에 표시된 각각의 범례는 아래에 제시한 네 가지 방법을 순서대로 이용하여 실험한 결과이다. 여기서 기존의 연구 방법인 VP-tree에 관해서는 한 개의 vp를 이용한 범위의 압축 방법보다는 복수 개의 vp를 이용한 방법 쪽이 더 유효하다는 연구 보고가 있기 때문에 본 실험에서는 후자인 VP-tree를 채택하였다.

- vp_all : 복수의 vp를 이용한 검색 범위의 압축 방법,
- vp_nn : 최근접점을 이용하여 검색 범위를 압축한 방법,
- vp_all_nn : vp_all과 vp_nn을 조합한 검색 범위의 압축법,
- AESA : AESA에 의한 검색 범위의 압축법 등 네 가지 이다.

우선, 각 차원에서의 거리 계산 횟수에 대한 실험 결과를 그림 6~그림 9에 나타내었다. 그림 6~그림 9은 각각 12, 24, 48, 96 차원 데이터의 실험 결과에 대한 내용이며, 가로축 k가 검색 건수, 세로축 calc_num가 거리 계산 횟수를 나타낸다. 그림에서 VP-tree을 보면 vp_all, vp_nn, vp_all_num의 순서로 거리 계산 횟수가 감소하고 있음을 알 수 있다. 한편 AESA는 VP-tree에 비해 적은 수의 거리 계산 횟수가 가능함을 알 수 있다.

다음으로 각 차원에서 검색 시간에 대한 실험 결과를 그림 10~그림 13에 나타내었다. 그림 10~그림 13는 각각 12, 24, 48, 96 차원 데이터에 대한 실험 결과이다. 그림 14는 AESA를 이용하여 모든 차원(12~96)에서의 데이터 검색 실행 시간을 나타내었다. 이들 그림의 가로축 k가 검색 건수이며, 세로축의 cpu-time는 초 단위로 표시하였다. 그림 10~그림 13의 VP-tree을 살펴보면 vp_all, vp_nn, vp_all_num 순으로 cpu-time이 감소하고 있다. 특히, vp_all와 vp_nn을 비교해 보면 그 차이가 미비하지만, vp_all_nn에 이르러서는 10% 정도의 향상이 눈에 보인다. 이는 최근접점을 이용한 검색 범위의 압축이 종래의 vp와는 다른 범위에서 처리되고 있으며, 쌍방(앞의 vp_all과 vp_nn을 조합한 방법)을 병용함으로서 효율적으로 검색 범위의 압축이 크게 좁혀졌기 때문이라 생각된다.

차원수 변화에 따른 실험 시간의 향상률에 관련하여 12차원에서 100 건의 검색을 실행하였을 때 얻어진 향상률이 5% 정도 이었던 것에 비해, 96 차원에서 100 건의 데이터를 검색하였을 때는 12%의 향상률을 얻을 수 있었다. 이와 같이, 본 논문의 방법을 이용하면 차원 수가 증가하여도 충분한 검색 범위의 압축 효과를 기대할 수 있다.

실험을 위해 구축한 색인 파일에 대한 상세 정보를 아래의 표 1에 제시하였다. dim은 차원 수, node는 노드 수, leaf_object는 리프 오브젝트의 수, index_size는 인덱스 데이터의 크기를 의미한다. 리프 노드에 있어서의 최대 분기점은 10으로 설정하였다. 또한 최근접점을 이용한 검색 범위의 압축 방법에 필요한 거리 리스트를 기록한 파일의 크기는 차원에

표 1. 색인 파일의 명세(specification)

dimension	node	leaf_object	index_size (byte)
12	2357	7643	6,000,640
24	2255	7745	5,742,592
48	2265	7735	5,767,168
96	2295	7705	5,844,992

표 2. 거리 목록 파일을 읽어 들인 회수의 비교

차원수	AESA	VP-tree
12	110	6
24	219	7
48	257	7
96	262	7

관계없이 313 MByte이었다.

다음으로 본 논문에서 제시하는 방법과 AESA를 비교하였을 때의 실험 결과를 설명하면 다음과 같다. AESA가 거리 계산 횟수 부분에서 VP-tree보다 우수함에도 불구하고, 검색 시간이 늦다는 것을 알 수 있었다. 그 이유는 거리 리스트 파일을 불러들이는 횟수의 차이를 생각해 볼 수 있다. VP-tree와 AESA가 거리 리스트 파일을 읽어 들이는 횟수를 표 2에 제시하였다. VP-tree의 거리 리스트 파일은 앞에서 설명한 리프 노드에 링크하는 리프 오브젝트와 그 밖의 모든 오브젝트와의 거리를 계산한 파일이다. 반면에 AESA의 거리 리스트 파일은 모든 오브젝트 사이의 거리를 계산한 파일이다.

AESA는 처리 과정을 반복할 때마다 거리 리스트 파일을 반드시 불러 와야 한다. 이 사실은 거리 계산 회수와 동일한 회수만큼 거리 리스트 파일을 반복하여 읽어 들여야 한다는 것이며, 이것이 검색 시간에 큰 영향을 주고 있음을 알 수 있다. 한편으로 VP-tree는 리프 노드의 리프 오브젝트의 검색 범위 압축에만 거리 리스트 파일을 읽어 들이면 되기 때문에 파일을 읽어 들이는 횟수를 극소수로 줄일 수 있다. 따라서 지금까지의 실험 결과를 종합하여 말하면 AESA 보다도 VP-tree 쪽이 검색 효율을 높이는 데에 유용하다고 할 수 있다.

5. 결 론

본 논문에서는 VP-tree의 리프 노드의 검색 알고리즘을 개량하여 리프 노드에서의 거리 계산 횟수를 삭감하고, 검색 속도의 향상에 기여하고자 하였다. 이 개선 방법을 이용하여 유사 영상 검색의 실험을 실행하였다. 그 결과 유사 영상 검색의 검색 시간을 5%~12% 줄일 수 있음을 알 수 있었다. 또한 VP-tree가 AESA 보다 검색 시간을 줄이는데, 더 유용하다는 사실도 알 수 있었다. 앞으로의 향후 과제에서는 보다 적은 색인의 크기로 거리 계산을 더 줄일 수 있는 검색 알고리즘을 고안하고자 한다.

참 고 문 헌

[1] Kita K., Tsuda, K., and Shishibori M., *Information Retrieval Algorithm,* Kyoritsu Shuppan, 2002.

[2] Yoshikawa M. and Uemura S., “Indexing Techniques for Multimedia Data,” *Journal of Information Processing Society of Japan,* Vol.42, No.10, pp. 953-957, 2001.

[3] Katayama N. and Satoh S., “Indexing Techniques for Similarity Retrieval,” *Journal of Information Processing Society of Japan,* Vol.42, No.10, pp. 958-963, 2001.

[4] Guttman A., “A Dynamic Index Structure for Spatial Searching,” *Proc. ACM SIGMOD '84,* pp. 47-57, 1984.

[5] Beckmann N., Kriegel H. -P., Schneider R., and Seeger B., “The R*-tree: An Efficient and Robust Access Method for Points and Rectangles,” *Proc. ACM SIGMOD*, pp. 322-331, 1990.

[6] White D.A. and Jain R., “Similarity Indexing with SS-tree,” *Proc. 12th Int. Conf. on Data Engineering*, pp. 516-523, 1996.

[7] Katayama N., and Satoh S., “SR-Tree : An Index Structure for Nearest Neighbor Searching of High - Dimensional Point Data,” *IEICE Transaction on Information and Systems,* Vol.J80-D-I, No.8, pp. 703-717, 1997.

[8] Berchtold S., Keim D. A., and Kriegel H. -P., “The X-tree An Index Structure for High Dimensional Data,” *Proc. 22nd VLDB*, pp. 28-39, 1996.

[9] Weber R., Schek H.J., and Blott S., “A Quantitative Analysis and Performance Study for Similarity - Search Methods in High-Dimensional Spaces,” *Proc. 24th VLDB*, pp. 194-205, 1998.

[10] 밧수리수브다, 고재필, “얼굴 인식을 위한 거리

척도 학습 방법 비교", 멀티미디어학회논문지, 제14권, 제6호, pp. 711-718, 2011.

[11] Ioka M., *A Method of Defining the Similarity of Images on the Basis of Color Information*, Technical Report RT-0030, IBM Tokyo Research Lab., 1989.

[12] Rubner Y., Tomasi C., and Guibas L.J., "The Earch Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval," *Proc. of the ARPA Image Understanding Workshop*, pp. 661-668, 1999.

[13] Ciaccia P., Patella M., and Zezula P., "M- tree: An Efficient Access Method for Similarity Search in Metric Spaces," *Proc. ACM SIGMOD Int. Conf. on the Management of Data*, pp. 71-79, 1995.

[14] Yianilos P. N., "Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces," *Proc. of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms(SODA '93)*, pp. 311-321, 1993.

[15] Fu A.W. -C., Chan P.M.S., Cheung Y.L., and Moon Y.S., "Dynamic VP-Tree Indexing for N-Nearest Neighbor Search Given Pair- Wise Distances," *The VLDB Journal,* Vol.9, No.2, pp. 154-173, 2000.

[16] Bozkaya T. and Ozsoyoglu M., "Distance-based Indexing for High Dimensional Metric Spaces," *Proc. of the 1997 ACM SIGMOD International Conference on Management of Data(SIGMOD '97),* pp. 357-368, 1997.

[17] Ishikawa M., Notoya J., Chen H., and Ohbo N., "A Metric Index MItree," *Transactions of Information Processing Society of Japan,* Vol.40, No.SIG6(TOD3), pp. 104-114, 1999.

[18] Vidal Ruiz, "An Algorithm for Finding Nearest Neighbours in (approximately) Constant Average Time," *Pattern Recognition Letters,* Vol.4, Iss.3, pp. 145-157, 1986.

[19] Iwasaki M., "Implementation and Evaluation of Metric Space Indices for Similarity Search", *Transactions of Information Processing Society of Japan,* Vol.40, No.SIG3(TOD1), pp. 24-33, 1999.

[20] Akama H., Konishi F., Yoshida T., Yamamuro M., and Kushima K., "External Key Search and Dynamic Data Insertion in Inverted File Indexing Method Applied for Nearest Neighbor Search," *Transactions of Information Processing Society of Japan,* Vol.40, No.SIG 8(TOD4), pp. 51-62, 1999.

[21] Corel Image, http://www.corel.co.kr/ 2011.

박 길 양

1997년 성신여자대학교 자연과학대학 수학과 학사
2004년 동국대학교 정보보호학과 석사
2008년 8월~현재 군산대학교 전자정보공학부 정보통신전파공학 박사과정
2008년 11월~현재 딕스비전(주) 대표 이사
관심 분야 : 디지털 영상 처리, 지능형 영상 보안, 컴퓨터 비전

이 상 곤

1998년 전북대학교 전산통계학과 석사
2001년 일본 국립 도쿠시마대학교 지능정보공학과 박사
2002년~현재 전주대학교 컴퓨터공학과 부교수

관심 분야 : 한국어 정보처리, 한글공학, 정보검색, 문서분류, 영상 처리 시스템의 구현

황 재 정

1986년 전북대학교 전자공학과 석사
1992년 전북대학교 전자공학과 박사
1992년 ~현재 군산대학교 전파공학과 교수

관심 분야 : 영상 통신 및 처리, 디지털 방송, 멀티미디어 스트리밍, 영상 보안, 인터넷 TV