2002 년 한국정보처리학회 춘계학술발표논문집 제 9 권 제 1 호

이미지 데이타 클러스터링을 이용한 검색 연구

김진옥, 황대준
성균관대학교 전기전자 및 컴퓨터공학부
e-mail : jinny@ece.skku.ac.kr

Study on the searching of images via clustering

JinOk Kim, DaeJoon Hwang
School of Electrical and Computer Engineering, SungKyunKwan University

요 약

이미지, 비디오, 오디오와 같은 멀티미디어 데이터들은 텍스트기반의 데이터에 비하여 대용량이고 비정형적인 특성을 가지기 때문에 검색이 어렵다. 또한 멀티미디어 데이터의 특징은 행렬이나 벡터의 형태로 표현되기 때문에 완전일치 검색이 아닌 유사 검색을 수행하여 사용자가 원하는 이미지와 유사한 이미지를 검색해야 한다. 본 연구에서는 멀티미디어 데이터 검색에 클러스터링과 인덱싱 기법을 같이 적용하여 유사한 이미지끼리는 인접 디스크에 클러스터하고 이 클러스터에 접근하는 인덱스를 구축하여 검색이 빠르게 이루어지는 유사 검색방법을 제안한다. 제안 검색 방법은 클러스터링을 생성하는 알고리즘과 해싱기법의 인덱싱을 같이 적용함으로써 VQ(Vector Quantization)보다 높은 재현율과 정확도를 보인다.

1. 서 론

내용 기반의 이미지 검색 기술[1,2,3]에서는 이미지를 고차원 특징벡터로 정규화 하여 고차원 공간의 근접지역에서 주어진 이미지의 벡터와 가장 유사한 벡터를 검색한다. 유사질의를 실행하기 위한 가장 단순한 방법은 모든 이미지를 읽고 질의 이미지까지의 거리를 계산하여 최근접 이미지를 선택한다. 하지만 이 방법의 실행시간은 데이터의 차원(D), 데이터집합의 크기(N), 알고리즘의 복잡도($O(ND)$)와 비례하기 때문에 검색공간을 줄임으로써 실행시간을 단축하는 트리구조(R-tree[6],R*-tree[7],SR-tree[8],K-d-b tree [9], TV-tree[10],X--tree[11],M-tree[12])가 제안되었다. 하지만 최근 연구결과[4,5]는 트리구조가 차원의 저주 문제를 야기시킨다는 것을 지적하고 있다. 근접 블록을 찾기 위해 인덱스구조를 왔다 갔다 하는 시간이 지나치게 많이 소요되며 유사 이미지를 찾는 과정에서 근접 블록 수가 데이터차원과 함께 지수적으로 커지기 때문에 성능은 역시 지수적으로 떨어진다는 것이다. 전통적인 트리구조는 데이터 차원이 높을 때 유사 오브젝트를 디스크에 랜덤적으로 분산 배치한다. 이에 따라 트리 인덱스구조는 데이터의 랜덤적 위치와 지수적으로 커지는 I/O 로 인해 고차원 공간에서 유사한 검색을 실행하는데 비효율적이다. 본 연구에서는 클러스터링과 인덱싱을 혼합한 새로운 유사도 검색 방법을 제안한다. 클러스터링과 인덱싱은 다양하게 연구되었지만 이미지 유사도 검색의 개별적인 방안으로 제안되고 있다.

클러스터링/인덱싱 스키마는 특징공간을 작은 셀 형태의 하위영역으로 나눈다. 나뉜 셀은 유사한 레코드를 클러스터링하고 클러스터를 인덱싱하는데 이용한다. 클러스터링 단계에서 각 셀에 위치한 이미지 수를 기록하고 이미지가 포함된 주변 셀을 같은 클러스터에 통합한다. 디스크 그리드의 해상도를 변경함으로써 클러스터를 다단계로 구축할 수도 있다. 즉 그리드가 더 섬세할수록 셀 크기는 작아지고 클러스터의 결과도 더 상세해 진다. I/O 의 효율성을 개선하기 위해 가장 촘촘하게 입도된 클러스터를 개별적인 연속적 파일로 저장하고 성글게 입도된 클러스터를 가능한 한 많이 촘촘한 클러스터와 같은 실린더 그룹에 저장한다. 클러스터를 형성한 후 알고리즘은 클러스터를 인덱싱하는 매핑 테이블을 만든다. 유사도 질의에 답하기 위해 먼저 질의 이미지의 특성 벡터를 해쉬하여 셀 ID 를 구한 다음 셀 ID 를 이용한 매핑 테이블로 질의 이미지가 위치한 클러스터를 탐색한다. 유사도 질의에 사용하는 I/O 의 수는 보통 두가지이다. 한가지는 클러스터를 찾는 I/O 이고 다른 한가지는 클러스터에서 읽는 연속적 파일의 I/O 이다 이와 같은 방법으로 클러스터링/인덱싱 스키마가 검색을 효율적으로 처리하여 찾고자 하는 이미지가 웹에 있으면 이미지를 포함하는 클러스터가 형성되어 이미지를 찾아낸다.

이 논문에서는 클러스터의 수와 크기에 영향을 미치는 계수를 제시하고 계수를 조율하는 절차를 설명하며 고차원 공간에서 빠른 검색을 지원하는 동시에 저장공간을 유지하는데 필요한 인덱스를 구축하는 방안을 제시한다. 데이터베이스 크기에 대해서는 선형적인 인덱스구조의 크기를 보존하는 힙 구조를 이용한다.

2. 관련연구

최근접 검색 기술이 고차원 공간에서 유사도 검색을 구현하는데 이용되어 왔다. 그러나 차원의 저주 때문에 최근 연구에서는 개략적으로 유사도 검색을 처리하는 방안을 제안하고 있다. White 와 Jain [4]은 질의 대상이 위치하는 데이터의 버켓만을 돌려주는 방법을 제안했다. 이 방법은 빠르지만 재현율(recall)은 낮다. Kleinberg[13]은 $O((D\log^2 D)+(N+\log^3 N))$ 과

$O(N + D\log^3 N)$ 시간 안에 돌아가는 두 가지 근접 알고리즘을 제안했다. 이 두 가지 다른 접근 알고리즘은 차원을 줄이고 병렬 자원을 사용하여 차원의 저주를 처리하는 방법을 제안했다. 이 방법들은 아주 높지 않은 고차원 응용에 적용할 수 있지만 고차원의 이미지 검색 문제를 해결하기는 어렵다.

QBIC[14], Virage[15]등 주어진 이미지와 유사한 이미지를 찾는 전통적인 내용기반의 이미지 검색 방법이 있지만 이 시스템들은 전통적인 트리 구조를 이용하여 구현되었으므로 차원의 저주 문제를 야기시킬 수 있다. CLARANS[16], BIRCH[17], DBSCAN[18], CLIQUE[19]와 같은 클러스터링 연구가 데이터베이스 분야에서 이루어지고 있다. 이 기술들은 대형 데이터집합에서 클러스터를 확인하는 데 높은 성공율을 보인다. 그러나 이 방법들은 데이터 검색과 추출을 효과적으로 처리하지 못한다. 본 논문의 클러스터링/인덱싱 접근은 디스크에 유사이미지를 클러스터링 함으로써 차원의 저주를 피한다 또한 클러스터링/인덱싱 접근이 유사도 검색을 대략적으로 수행하지만 데이터 클러스터을 통해 재현율을 높인다.

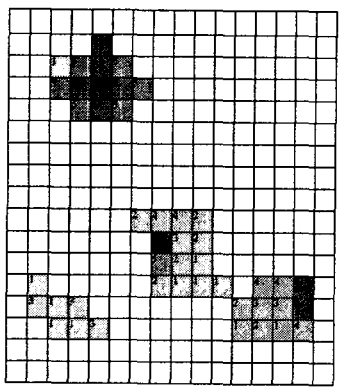
3. 클러스터링과 인덱싱

이 장에서는 클러스터링과 인덱싱 스키마를 설명한다. 스키마의 주요 아이디어는 유사 오브젝트를 추출하기 위한 디스크 지연도(latency) 를 최소화하도록 디스크에 유사 데이타끼리 클러스터하고 클러스터를 인덱스 하는데 해싱을 이용하여 클러스터 탐색 비용을 최소화한다. 접근방법은 세가지 단계로 구성된다.

1) i 번째 차원을 2^k 개 영역으로 나누고 최소 비트수를 사용하여 특성벡터가 속한 셀을 인코딩한다.
2) 클러스터내로 셀을 통합한다. 셀은 각각 다른 모양의 클러스터를 구축하기 위한 최소 블록으로서 차원을 나눈 하위영역이 작을수록 셀은 더 작아진다. 클러스터는 연속적인 파일형태로 디스크에 저장된다.
3) 클러스터를 참조하는 인덱스 구조를 만든다. 셀은 최소 주소단위이다. 인코딩 스키마는 오브젝트를 셀 ID 로 해쉬하고 I/O 1 회에 속하는 클러스터를 추출한다

3.1 클러스터링 방법

고차원 공간에서 클러스터링을 효율적으로 수행하기 위해 클러스터 구성 알고리즘을 제안한다. 그림 1 은 2D 상에서 같은 크기로 나뉜 그리드에 분포한 포인트(오브젝트)를 클러스터링한 것이다.



(그림 1. 클러스터링 생성 방법)

클러스터 구성 알고리즘은 다음 방법으로 이루어진다

1)클러스터 구성 알고리즘은 먼저 각 셀의 높이(오브젝트 수)를 기록한다.
2)클러스터 구성 알고리즘은 은 포인트 집적도가 가장 높은 셀에서 시작한다. 이 셀은 최초 클러스터의 중심점이다.(그림 1 에서는 7 로 표시된 셀에서 시작한다)
3)클러스터 구성 알고리즘은 한번에 한 단위씩 내려온다. 셀은 세 가지 조건 중의 하나인데 어떤 클러스터에도 인접하지 않거나 오직 한 클러스터에만 인접하거나 한 개 이상의 클러스터에 인접해 있다.클러스터 구성 알고리즘은 취하는 방법은
(a)만약 셀이 어떤 클러스터에도 인접하지 않으면 셀은 새로운 클러스터를 생성하는중심이 된다.
(b)셀이 한 개 클러스터에만 인접하면 셀을 클러스터에 조인한다.
(c)셀이 한 개 이상의 클러스터에 인접하면 클러스터 결정 알고리즘은 어떤 클러스터에 셀이 속하는지 또는 셀이 속한 클러스터끼리 통합되는 것을 결정한다.

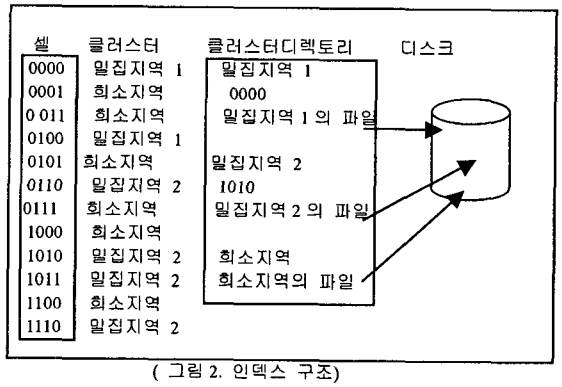
4)클러스터 구성 알고리즘은 셀의 높이가 임계값이 되면 끝난다. 어떤 클러스터에도 속하지 않는 셀은 외곽 클러스터에 통합되고 한 개의 연속적 파일로 저장된다.

3.2 클러스터 인덱싱

클러스터를 형성한 후의 마지막 단계는 클러스터에 빨리 접근할 수 있는 인덱스를 구축하는 것이다. I/O 의 효율성을 개선하기 위해 가장 촘촘하게 압도된 밀집 클러스터를 개별적인 연속적 파일로 저장하고 성글게 압도된 희소 클러스터를 가능한 한 많이 촘촘한 클러스터와 같은 실린더 그룹에 저장한 후 오브젝트를 클러스터에 사상하기 위해 사상 테이블을 생성하고 클러스터에 대한 정보를 기록하기 위해 클러스터 디렉토리를 구축한다.

사상 테이블의 크기는 힙의 크기에 비례한다. 힙은 각 셀에 대한 정보를 기록한다. 최악의 경우 각 포인트에 대해 셀을 한 개씩 갖게 되어 N 개의 셀 구조가 된다.

각 셀 구조는 셀 ID, 포인트 수, 클러스터 ID, 힙 유지를 위한 포인터로 이루어지며 사상 테이블은 셀 ID 와 클러스터 ID 로 구성된다. 포인트 수, 클러스터 ID, 포인터에 대한 스토리지 소요량은 불변하며 셀 ID 의 크기는 데이터 차원 D 에 비례한다. 힙의 전체 소요 스토리지는 $O(ND)$ 정도이다. 본 논문의 제안 알고리즘은 트리 유사 인덱스 구조와는 달리 검색 시 메모리에 셀을 포함하고 있는 블록을 읽을 때 외에는 메모리 용량을 크게 필요로 하지 않는다. 테이블의 디스크 스토리지 소요는 트리 유사 인덱스 구조와 비교 가능하다. 사상 테이블 외 두 번째 필요한 인덱스 구조는 클러스터 디렉토리이다. 클러스터 디렉토리는 각 클러스터 해당 ID 와 클러스터가 오브젝트를 저장한 파일의 이름, 최고 높이의 포인트 셀인 클러스터 중심 셀 ID를 가지고 있다.



(그림 2. 인덱스 구조)

3.3 제어계수

클러스터의 입도(Granularity)는 네 가지 계수로 제어한다

- D : 데이터차원 또는 오브젝트 특징
- b : 각 차원을 인코딩하는데 사용하는 비트의 수
- N : 오브젝트의 수
- θ : 수평계수

셀의 수는 D, b 계수로 결정되며 $2^{D\times b}$ 로 쓴다. 셀의 평균 포인트 수는 $\frac{N}{2^{Db}}$ 이다. 두 인자는 원하는 포인트 밀도를 결정한다. 낮은 포인트 밀도는 많은 수의 셀을 만들어 내면서 많은 수의 작은 클러스터들을 생성하기 때문에 좋지 않다. 반면 높은 포인트 밀도는 클러스터 수는 적지만 대형 크기의 클러스터를 만들어 내기 때문에

역시 좋지 않다. b 값의 변화는 그리드의 해상도에 영향을 미치고 클러스터의 수와 크기도 결정한다. θ 값 역시 클러스터의 크기와 수에 영향을 미친다. 임계 값이 낮아지면 포인트 밀집 영역의 수와 크기는 증가하는 대신 포인트가 없는 희소영역의 크기는 감소한다. 희소영역이 감소하면 I/O 의 효율성을 높일 수 있지만 밀집영역의 수와 크기가 증가해 포인트 밀집영역 클러스터들의 경계선이 불분명해 진다. 그래서 적절한 데이터 분포와 원하는 클러스터 크기를 고려하여 b 와 θ 의 값을 조절해야 한다.

3.4 알고리즘의 시간 복잡도

N 은 데이터집합에서 오브젝트 수이고 M 은 채워진 셀의 수이다. 클러스터 형성 알고리즘의 시간 복잡도는 오브젝트의 셀 ID 를 계산하는데 $O(D)$ 가 걸리고 두개의 셀이 인접하는지 확인하는데 $O(D)$ 가 걸린다고 하면 첫번째 단계에서 셀 ID 와 높이를 유지하는데 힙을 이용하는데 셀 ID 가 주어지면 셀이 힙에 위치하는지 확인하는 시간이 $O(D\times\log M)$ 이 걸린다 그래서 클러스터 형성 알고리즘의 첫번째 단계의 시간 복잡도는 $O(N\times D\times\log M)$ 이다. 두번째 단계에서 인접한 셀을 찾는 시간은 각 차원에서 주어진 셀은 적어도 3 개의 하위영역과 인접하기 때문에 $O(\min\{3^D,M\})$ 이다. 2 개 고차원공간에서 $M\ll 3^D$ 이기 때문에 두번째 단계의 시간 복잡도는 $O(D\times M^2)$ 이다. 그래서 총체적인 시간 복잡도는 $O(N\times D\times\log M)+O(D\times M^2)$ 가 된다.

4. 평가

실험에서는 주어진 질의에 대해 가장 유사한 k 개 오브젝트를 찾아내거나 k 최근접 오브젝트를 찾아낸다. 10 개의 최근접 데이터를 찾기 위해 먼저 데이터집합을 스캔한다. 질의 결과는 세 종류의 결과치로 확인한다.

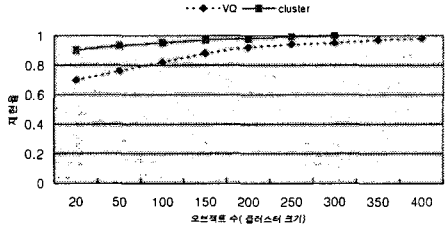
- X 번의 I/O 후 재현율: X 번의 I/O 가 이루어진 후 n 개의 결과를 추출
- X 번의 I/O 후 정확도: X 번의 I/O 가 이루어진 후 n 개 추출 오브젝트의 정확도

본 논문에서는 재현율과 정확도로 알고리즘을 평가한다. 평가를 수행하기 위해 이미지 CD 를 이용하여 3,000 개 이미지 데이터베이스를 구축했다. 검색을 수행하기 위해 세 단계를 거쳐 이미지의 특징벡터를 추출한다. 이미지 특징벡터 추출과정은 내용기반 이미지 검색시스템에서 주로 적용하는 Daubeches Wavelet transform 을 적용한다. 먼저 정규크기와 형식으로 이미지를 바꾼다. 그리고 이미지에 Daubeches Wavelet transform 을 적용하여 계수를 추출한다. 마지막으로 이미지의 특징벡터로 선택한 HSV 영도, 채도, 색도의 세가지 색상 특징치, 모멘트, 대비, 연관성, 분산, 혼잡도의 5 가지 질감 특징치, 영역 크기 정보, 푸리에 기술자를 이용한 모양 및 위치정보 특징치를 특징공간에 48 차원의 웨이블릿 계수로 저장한다. 그런 다음 데이터집합을 적절하게 클러스터링하기 위해 클러스터링 알고리즘을 적용하고 데이터집합에서 패턴을 찾도록 d 와 θ 계수를 조정하여 2 를 d 에 적용하고 1 을 θ 에 적용하면 평균 16 개 이미지 수를 갖는 25 개 클러스터를 찾는다. 대부분 유사 이미지는 같은 클러스터에 위치한다. 질의 이미지가 클러스터 중심에 있을때는 검색이 잘되지만 포인트 밀집 지역의 주변이나 희소지역에 있을 때는 잘 감지되지 않는다. 이 경우에는 질의 이미지의 주변 포인트들을 근접 클러스터로 별도 분류해 검색한다.

제안 알고리즘의 비교 성능 평가를 위해 VQ(Vector Quantization)리즘[23]에 데이터 특성벡터를 적용시켰다. VQ(Vector Quantization)알고리즘은 고차원 공간에서 데이터를 클러스터링 하기 위해 구현된 $k-$Means 유사 알고리즘이다. VQ(Vector Quantization)은 내용기반의 이미지 추출을 위해 고차원 공간에서의 인덱싱과 압축에 많이 쓰인다. 재현율을 측정하기 위해 클로스 밸리데이션 기법을 적용하여 10 회의 테스트를 했다. 각 회에서 30 개의 테스트 이미지를 이용하여 제안 알고리즘과 VQ 로 데이터베이스를 검색하였다. 30 개 이미지 질의의 재현율로 매회 평균 질의 재현율을 구한다음 구한 재현율을 평균했다. 재현율은 클러스터링 알고리즘과 클러스터 크기를 고려했다.

4.1 재현율

실험을 통해 그림 3 과 같은 결과를 볼 수 있다.
25 개의 클러스터로 나뉜 데이터집합에서 5 회 I/O 에 클러스터 구성 알고리즘은 약 84%의 재현율에 해당하는 오브젝트를 검색한다. 5 회를 더 테스트한 후 재현율은 98%까지 높아졌다. 클러스터를 읽으면 읽을수록 재현율은 높아지지만 처리 속도는 느려진다.
15 개의 클러스터를 읽은 후 재현율은 100%에 가까워지면서 10 개의 최고 유사 오브젝트를 찾는데 데이터집합에서 16%(4/25)의 데이터를 읽는다. VQ 는 5 회 I/O 에 69%의 재현율을 보이고 5 회를 더한 I/O 에 90% 정도의 재현율을 보였다



(그림 3. 클러스터 크기 대 재현율/5 회의 I/O)

클러스터링/인덱싱은 VQ 보다 높은 재현율을 보이며 클러스터의 모양과 크기에 대해 더 유연하다. VQ 는 선형함수로 클러스터를 바운드하는 반면 클러스터링은 여기에 제약을 받지 않는다

4.2 클러스터 크기에 따른 재현율

클러스터에서 평균 오브젝트 수를 클러스터 크기로 규정하고 재현율에 클러스터 크기가 어떤 영향을 미치는지 보기 위해 각각 다른 클러스터 크기로 하여 재현율 값을 측정했다
클러스터링의 클러스터 크기를 결정하기 위해 d , θ 값을 정한다. 클러스터 크기를 50 개에서 400 개 오브젝트로 규정하여 실험했다. 그림 3 은 5 회의 I/O 를 행한 후 다른 클러스터 크기를 적용했을 때의 재현율이다. 클러스터 구성 알고리즘이 VQ 보다 높은 재현율을 보인다.

4.3 최근접 오브젝트 수 대비 정확도

그림 4 는 찾아낸 10 개의 최근접 오브젝트 대비 추출 오브젝트 수로 계산한 정확도를 보여준다. 클러스터 구성 알고리즘이 VQ 보다 높은 정확도를 보인다

	재현율	90%	92%	96%	98%	100%
	최근접 오브젝트수	6	7	8	9	10
VQ	추출 오브젝트 수	180	220	380	520	1,120
	정확도	3.3%	3.2%	2.1%	1.7%	0.9%
cluster	추출 오브젝트 수	61	83	120	250	310
	정확도	9.8%	8.4%	6.7%	3.6%	3.2%

(그림 4. 10 개 최근접 오브젝트에 대한 정확도)

5 결론

본 논문에서는 고차원 공간에서 차원의 저주 문제를 피하면서 유사도 검색을 수행하는 새로운 방법으로 데이터를 클러스터링하고 이 클러스터들을 검색하는 방안을 제안했다. 이 방법은 유사한 오브젝트를 디스크에 클러스터링하고 질의 오브젝트 근처의 클러스터를 찾음으로써 유사도 검색을 수행하는 방식으로 디스크로부터 관련 정보를 연속적으로 추출하고 클러스터링 함으로써 I/O 시간을 줄이고 연관된 클러스터를 연속적으로 검색한다. 또한 클러스터의 중심 정보를 이용하여 인접 영역을 효과적으로 검색하기 때문에 패턴을 클러스터링하는 고차원 데이터집합에 적용가능하며 특히 동종의 데이터집합 검색에 효율적인 방안이다.

이 연구의 주된 연구결과를 요약하면 다음과 같다.

- 스토리지 클러스터에 데이터 클러스터를 사상하는 방안을 제안하고 이렇게 만들어진 클러스터에 접근하는 인덱스를 구축

하였다

- 클러스터를 형성하는 클러스터 결정 알고리즘을 제안했다.
- 질의 이미지에 클러스터를 연결시키는 알고리즘을 이용하여 유사도 질의를 효과적으로 처리하는 방안을 제시했다.

참조

[1] P. Aigrain, H. Zang and D. Petkovic."content based Representaiton and Retrieval of Visual Media: A State-of-the-Art Review" Multimedia Tools and Applications, vol3, pp179-202,1996

[2] T.Hermes,C.Klauck,J.Krey and J, Zhang"Image Retrieval for information systems" in Proc. SPIE Vol24(20): Storage and Retrieval for Image and Video Databases" pp394-405,February 1995

[3] Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In Proceeding of International Conference Visual Information System, 1999.

[4] D. A. White and R. Jain. Algorithms and strategies for similarity retrieval. Stanford Technical Report, August, 1998.

[5] R. Weber, H. Schek and S. Blott. A quantitative anlaysis and performance study for silimarity-search methods in high-dimensional spaces. Proceedings of the 24th VLDB, pages 194-205,1998

[6] A. Guttman. R-tree; a dynamic index structure for spatial searching. Proceedings of ACM Sigmod. June 1984

[7] N. Beckmann. H. P. Kriegel, R. Schneider and B. Seeger. The r*-tree: an efficient and robust acces method for points and rectangles. Proceedings of ACM Sigmod May 1990.

[8] N. Katayama and S. Sotoh. The sr-tree: An index structure for high-dimensional nearest neighbor queries. Proceedings of ACM SIGMOD, May 1997.

[9] J. T. Robinson. The k-d-b-tree: A search structure for large multidimensional dynamic indexes. Proceedings of ACM SIGMOD, April, 1981.

[10] K. L. Lin, H. V. Jagadish and C. Faloutsos. The tv-tree: an index structure for high-dimensional data. VLDB Journal,3(4),1994

[11] S. Berchtold. The x-tree_ An index structure for high demensional data. Proceedigns of the 22nd VLDB, August 1996.

[12 P. Ciaccia, M. Patella and P. Zezula, M-tree: An efficient access method for similarity search in metric spaces. Proceedings of the 23rd VLDB, August 1997.

[13] J. M. Kleinberg. Two algorithms for nearest-neighbor search in hgh dimensions. Proc. Of 29th ACM Symposium on Theory of Computing, 1997.

[14] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, and et al. Query by image and video content: the QBIC system. IEEE Computer, 28(9):23-32,1995.

[15] A. Hampapur, A. Gupta, B. Horowitz, C. Fuller, J. R. Bach M. Gorkani, and R. C. Jain. Virage Inc., Virage Video Engine. In Proc. SPIE Vol. 3022 : Storage and Retrieval for Image and Video Databases. pp.188-198, Feb. 1997.

[16] R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. Proceedings of the 20th VLDB, September, 1994.

[17]T. Zhang,R. Ramakrishnan and M. Liny. Birch: An efficient data clustering method for very large databases. Proceedings of Sigmod, June 1996.

[18] M. Ester, H. P. Kriegel, J. Sander and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining, August 1996.

[19] R.Agrawal, J.Gehrke, D.Gunupolus and P. Raghavan. Automatic subspace clustering of high diemensional data for data mining applications. Proceedings of ACM sigmod, June 1998

[20] G. Lu and A.Sajjanhar"Region based shape representation and similarity measure suitable for content based image retrieval" Multimedia systems, pp165-174,1999

[21] 장동식,정세환,유헌우,손용준,"VQ 를 이용한 영상의 객체 특징 추출과 이를 이용한 내용기반 영상 검색" 정보과학회 논문지, 컴퓨팅의 실제 제 7 권 제 6 호, pp 724-732, 20001.

[22] R. Weber, H. Schek and S. Blott. A quantitative anlaysis and performance study for silimarity-search methods in high-dimensional spaces. Proceedings of the 24th VLDB, pages 194-205,1998

[23] A.Gersho and R. Gray. Vector Quantization and Signal Compression. Kluwer Academic,1991..