필터링 기법을 이용한 고차원 색인 기법의 설계 및 구현

한성근, 장재우
전북대학교 컴퓨터공학과

Design and implementation of high-dimensional indexing scheme using filtering method

Sung-Geun Han, Jae-Woo Chang
Dept. of Computer Engineering, Chonbuk National University

요 약

현재 멀티미디어 응용 분야에서 고차원 데이터에 대한 색인 기법이 아주 중요시 되고 있다. 특히, 인터넷의 보급으로 멀티미디어 정보에 대한 수요가 급증함에 따라 멀티미디어 객체에 대한 효율적인 색인 기술이 절실히 필요하게 되었다. 멀티미디어 객체들은 특징 벡터들로 표현이 되며, 대부분 고차원 특징 벡터를 형성하게 된다. 이러한 고차원 특징 벡터를 색인 및 검색하기 위하여 다양한 방법들이 제시되었다. 그러나, 차원이 증가할수록 검색 성능이 급격히 저하되는 dimensional curse 문제를 완전히 해결하지는 못했다. 본 논문에서는 필터링(filtering) 기법을 사용하여 개선된 고차원 색인 기법을 설계 및 구현한다.

1.서론

최근 들어 컴퓨터 기술의 발달과 인터넷의 확산으로 인하여 누구나 쉽게 멀티미디어 자료를 이용할 수 있게 됨으로써 멀티미디어 데이터에 대한 요구가 급증하고 있다. 이를 위해 멀티미디어 데이터베이스를 기반으로 하는 정보 서비스가 대두되고 있으며, 효과적인 내용-기반 멀티미디어 검색 기법이 요구되고 있다. 내용-기반 멀티미디어 정보 검색을 위해 사용자는 멀티미디어 객체로부터 n(>1)개의 특징 벡터(feature vector)를 추출하여 데이터베이스에 저장한 다음, 이를 이용하여 사용자가 검색하려고 하는 멀티미디어 객체의 특성 값을 이용하여 이것과 가장 유사한 값을 갖는 멀티미디어 객체를 찾게 된다.

내용-기반 멀티미디어 정보 검색을 위한 고차원 색인 구조에 기반한 사용자의 질의 형태는 크게 3 가지로 나누어진다. 첫째, 주어진 질의와 정확하게 일치하는 객체를 검색하는 포인트 질의(point query), 둘째, 주어진 질의에 대한 임의의 유사성 범위 내에 포함되는 객체를 검색하는 범위 질의(range query), 마지막으로 주어진 질의 객체와 가장 유사한 객체 k 개를 찾는 k-최근접 탐색 질의(k-nearest neighbor search query)이다. 객체간의 유사성은 일반적으로 특징벡터 공간에서 유클리디언 거리를 이용한다. 이러한 질의 형태는 내용-기반 검색에서 매우 중요하며 고차원 색인 구조는 이를 효과적으로 제공하기 위한 형태를 취하고 있다. 하지만 멀티미디어 객체로부터 추출된 특징벡터의 차원이 증가함에 따라 기존의 색인 구조의 검색 성능은 급격히 저하된다[1]. 차원 증가에 따른 경계 효과(boundary effect), 유사성 측정의 차원에 대한 지수 함수적인 의존성, 고차원 특징 벡터 공간의 기학학적 표현의 불가능으로 인한 직관력 상실로 'dimensional curse'를 해결하는 최적의 방법을 찾는데 어려움을 겪고 있다. 이를 해결하기 위한 많은 연구들이 진행되고 있으나 아직까지는 미흡한 실정이다.

본 논문에서는 차원이 증가할수록 나타나는 이러한 문제들을 해결하기 위한 방법으로 시그니처를 이용한 필터링 기법을 제안한다. 객체들의 특징 벡터의 시그니처를 사용하여 데이터를 클러스터링한 후, 실제 객체의 특징 벡터를 접근하기 이전에 필터링함으로써 검색 성능을 향상시킬 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련 연구로 기존의 다양한 고차원 색인 기법들을 살펴 보고, 3 장에서는 제안하는 필터링을 이용한 새로운 고차원 색인 기법에 대해 소개한다. 4 장에서는 제안한 방법의 실험을 통해 성능을 평가하고, 5 장에서는 결론을 내린다.

2. 고차원 색인 기법

고차원 특징벡터에 적합한 색인 구조를 설계하기 위해서 내용-기반 이미지 색인 기법으로 연구된 기존의 고차원 색인기법들에는 k-d-트리, VAMSplit k-d-트리, SS-트리, Pyramid-트리, TV-트리, X-트리 등이 있다[2,3].

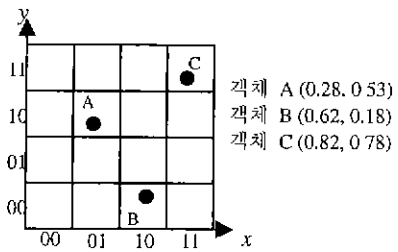
k-d-트리는 데이터가 거의 변경되지 않는 정적 환경에서 적합하도록 제안된 방법으로, 모든 차원에 대한 노드의 팬-아웃

(fan-out)을 일정하게 유지하여 삽입 시간이 빠르고, 겹침 영역이 발생하지 않는 장점을 가지지만, 검색 성능이 데이터의 삽입 순서에 매우 의존적이며, 많은 빈공간(dead-space)로 인해 검색 효율이 떨어지는 단점이 있다. VAMSplit k-d-트리는 k-d-트리의 분할 알고리즘을 개선한 방법이지만 모든 데이터를 메모리에 상주시켜서 작업해야 하는 문제점을 갖고 있다. SS-트리는 구 영역(spherical region)으로 데이터 공간을 분할하는 방법으로 유사성에 기반한 검색을 용이하게 하는 반면, 겹침 영역이 많이 발생함으로써 검색 성능이 저하된다. TV-트리는 차원이 증가함에 따라 트리 노드의 팬-아웃(fan-out)이 급격히 작아짐으로써 성능이 급격히 저하되는 문제점을 해결하기 위해 제안되었는데, 루트 노드에 근접한 노드들에 대해서는 단지 몇 개의 차원만을 사용해서 높은 팬-아웃을 갖고, 트리의 깊이가 깊어질수록 보다 많은 차원들을 사용함으로써 더욱 노드들의 분별력을 높이도록 하였다. 따라서, 특징 벡터의 차원이 증가할수록 검색 속도와 저장 공간의 요구량이 급속하게 증가되는 문제점을 해결하였다. 하지만, 중요도에 따른 우선 순위가 부여되어야 하는 가정과 특징값들의 차원과 정확히 일치해야 하는 가정을 전제로 하고 있으므로 이를 만족시켜야 하는 문제를 갖는다. 마지막으로, X-트리는 R-트리가 차원이 증가함에 따라 겹침 영역이 증가하여 검색 성능이 현저히 저하되는 문제점을 방지하기 위해 제안된 방법이다. 이를 위해 디렉토리에서의 겹침 영역을 피하기 위한 분할 알고리즘과 수퍼 노드 개념을 이용한다. 하지만, 여전히 차원이 증가하면 검색 성능이 현저히 저하되는 문제점을 내포하고 있다.

3. 필터링을 이용한 새로운 고차원 색인 기법

3.1 특징 벡터의 시그니쳐 변환

고차원 특징 벡터에 대한 시그니쳐는 특징 벡터 각각의 차원에 따른 요약들의 집합이므로, 각 차원에 해당하는 벡터의 성질을 유지할 수 있어야 한다. 또한, 질의를 처리하기 위해서, 질의 벡터와 시그니쳐들 사이의 관계를 표현할 수 있어야 한다. 따라서, 특징 벡터를 시그니쳐로 변환하기 위해 데이터 공간을 셀(cell) 단위로 나누고, 셀에 대한 시그니쳐를 사용한다 [2,4] 이렇게 만들어진 시그니쳐는 각각의 셀을 대표하는 값이 되며, 셀에 대한 정보를 포함하게 된다 [그림 1]은 2 비트 시그니쳐를 사용할 경우, 객체 A, B, C에 대하여 2 차원 공간에서 시그니쳐를 생성하는 과정을 보여 준다.



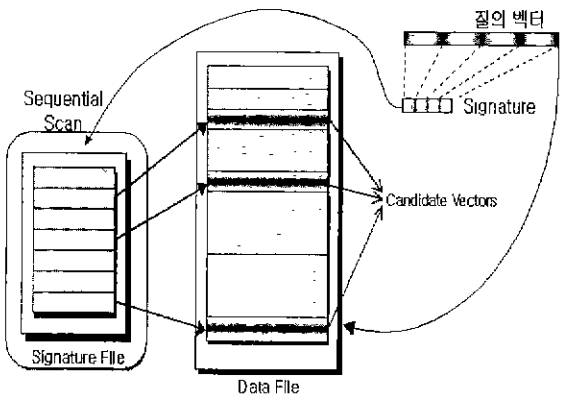
[그림 1] 2 차원 공간에서 시그니쳐를 만들어내는 과정

3.2 스캔-기반 필터링(Scan-based Filtering) 기법

제안한 시그니쳐는 셀에 대한 요약 표현이며, 하나의 셀 안에는 셀의 범위 내에 포함되는 여러 개의 객체가 포함될 수 있다 즉, 비슷한 성질을 가지는 객체가 클러스터링(clustering)되는 특징이 있다. 따라서, 어떤 데이터를 검색하는 경우, 직접 실제 객체를 접근하기 전에, 먼저 시그니쳐로 표현되는 셀을 접근하여 필요한 셀만을 선택하고, 선택된 셀에 대해서만 직접 실제 객체를 접근함으로써, 시그니쳐에 의한 필터링 효과를 얻을 수 있게 된다 스캔-기반 필터링 기법은 실제 객체의 전체 집합을 순차적으로 탐색하기 전에, 시그니쳐를 순차 탐색하는 방법이다.

X-trec 와 같은 고차원 색인 기법은 차원이 높아짐에 따라 성능이 저하되는데, 실제로 16 차원 이상이 되면 데이터베이스에 저장된 모든 객체들의 특징 벡터들을 순차 탐색하는 방법보다도 성능이 떨어지게 된다[1]. 따라서, 스캔-기반 필터링 기법은 이러한 순차 탐색을 이용하는 방법이다. 그러나 이 방법에서는 실제 특징 벡터들에 대한 순차 탐색을 수행하기 전에 각각의 특징 벡터에 대한 시그니쳐 탐색을 수행하여 필터링함으로써, 탐색 성능을 향상시키게 된다. 차원이 크면 클수록, 객체에 대한 특징 벡터의 전체 데이터 크기 또한 늘어나게 된다. 따라서, 하나의 버퍼를 사용할 경우, 버퍼 안에 로딩할 수 있는 객체의 특징 벡터 수가 줄어들게 되며, 순차 탐색을 할 경우 그만큼 디스크 I/O 수가 많이 필요하게 된다. 그러나, 시그니쳐를 사용하게 되면, 실제 특징 벡터에 비해 시그니쳐의 데이터 크기가 상대적으로 작기 때문에 버퍼에 로딩할 수 있는 객체의 시그니쳐 수가 많게 되고, 디스크 I/O 수를 줄이게 되어 검색 성능을 향상할 수 있게 된다.

스캔-기반 필터링 기법에서는 두 개의 파일을 사용하고 있는데, 객체의 특징 벡터들을 저장하고 있는 데이터 파일과 벡터에 대한 시그니쳐를 저장하는 시그니쳐 파일이다. 새로운 객체의 특징 벡터를 저장하는 경우, 우선 특징 벡터를 시그니쳐로 변환하고, 변환된 시그니쳐를 시그니쳐 파일에 저장한다. 그런 다음 실제 특징 벡터를 데이터 파일에 저장하게 된다. 이 때, 저장된 시그니쳐와 특징 벡터는 각각의 파일에서 똑같



[그림 2] 스캔-기반 필터링 기법에서의 검색 방법

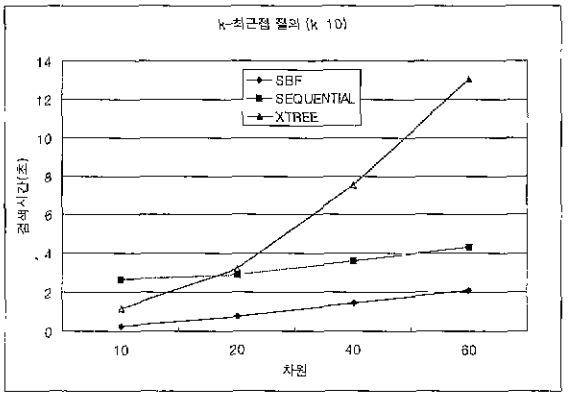
은 인덱스 위치값을 갖게 함으로써, 검색시에 이 값을 이용할 수 있다.

[그림 2]는 이렇게 생성된 두 개의 파일에 대해 질의를 처리하는 과정을 보여 준다 질의가 주어지면, 우선 질의 벡터에 대한 시그니처 변환을 수행하여 질의 시그니처를 생성한다 시그니처를 통해 각 셀의 범위값을 구할 수 있기 때문에, 이 값을 이용하여 질의 벡터의 검색 범위 안에 있는 시그니처들만을 다음의 검색 대상이 되는 후보 시그니처로 선택하게 된다. 즉, 시그니처에 의한 필터링이 수행되게 된다 그런 다음, 시그니처 탐색을 통해 선택된 각 후보 시그니처의 실제 특징 벡터들을 데이터 파일에서 순차 탐색함으로써 최종적으로 주어진 질의를 처리하게 된다. 이 때, 선택된 후보 시그니처들의 시그니처 파일 내의 인덱스 값과 데이터 파일 내의 후보 시그니처가 가리키는 특징 벡터의 인덱스 값이 같기 때문에, 후보 시그니처에 대한 특징 벡터의 위치를 데이터 파일에서 쉽게 찾아낼 수 있어서, 부가적인 다른 연산이 필요가 없게 된다. 시그니처에 의한 필터링 효과가 클수록 검색 성능은 우수하게 된다.

4. 실험 및 성능 평가

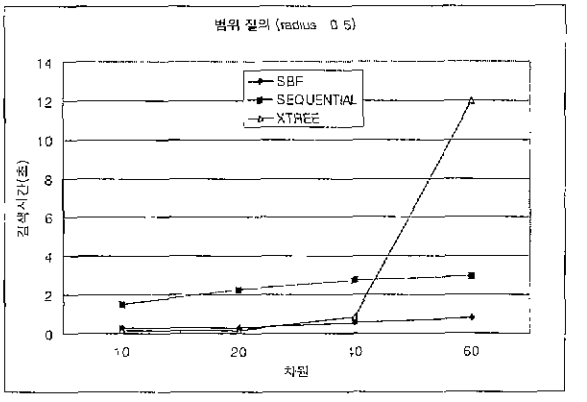
제안한 방법은 CPU 450MHz (dual), 메모리 128MB 의 Windows NT 4.0 에서 MS-Visual C++ 5.0 을 사용하여 구현하였다. 실험에 사용된 데이터는 X-트리에서 사용한 랜덤 데이터 생성 함수를 이용하여 10, 20. 40, 60 차원의 데이터 100.000 개를 사용하였다 제안한 방법(SBF)과의 성능 비교는 실제 객체를 순차 탐색하는 방법(SEQUENTIAL)과 현재 고차원 색인 기법에서 우수한 성능을 나타내는 X-트리(XTREE)를 비교하였다 또한, 비교 대상 질의는 k 값이 10 일 때의 k-최근접 질의와 radius 값이 0.5 일 때의 범위 질의를 사용하여 성능 평가를 수행하였다

[그림 3]은 k 값이 10 일 때, 각 차원에 따른 k-최근접 질의에 대한 검색 성능을 나타내며. [그림 4]는 radius 값이 0.5 일 때, 범위 질의에 대한 검색 성능을 나타낸다.



[그림 3] k-최근접 질의에 대한 성능 비교

그림에서 알 수 있듯이 제안한 방법은 차원이 증가함에 따라 급격히 성능이 저하되는 문제를 해결할 수 있었으며, 다른 두 방법보다 검색 성능이 우수함을 알 수 있다



[그림 4] 범위 질의에 대한 성능 비교

5. 결론

본 논문에서는 차원이 증가할 때 검색 성능이 급격히 저하되는 기존의 고차원 색인 기법의 dimensional curse 문제를 해결하기 위해 필터링 기법을 이용한 새로운 고차원 색인 기법을 제안하고, 설계및 구현하였다 제안한 방법에서는 객체의 특징 벡터에 대한 시그니처 정보를 유지함으로써, 특징 벡터를 직접 접근하기 전에 특징 벡터에 대한 시그니처를 먼저 접근함으로써, 필터링을 수행하여 검색 성능이 기존의 방법보다 향상됨을 실험을 통해 알 수 있었다. 향후 연구 계획으로는 셀(cell) 단위의 시그니처에 대해 좀 더 효율적인 필터링을 수행하는 방법을 찾아내는 것이다

참고 문헌

[1] Berchtold S., Bohm C., Keim D., Kriegel H -P, "A Cost Model for Nearest Neighbor Search in High-Dimensional Data Space", ACM PODS Symposium on Principles of Databases Systems, Tucson, Arizona, 1997.

[2] Stefan Berchtold, Daniel A. Keim "High-Dimensional Index Structures, Database Support for Next Decade's Applications (Tutorial)", SIGMOD Conference, 1998.

[3] 최질성, 유재수, 양재동, "내용 기반 이미지 검색 시스템에 관한 연구", 한국정보과학회 데이타베이스 연구회지 12 권 4 호 pp.97-116. 1996

[4] Roger Weber, Stephen Blott, " An Approximattion-Based Data Structure for Similarity Search", Technical report Nr. 24, ESPRIT project HERMES (no. 9141), October 1997.