

MIDAS-III에서 내용기반 이미지 검색을 위한 CIR-트리 관리기의 설계 및 구현

이희종*, 송석일*, 이석희*, 유재수*, 조기형*, 이훈순**, 이장선**

*충북대학교 정보통신공학과

** 전자통신연구원 컴퓨터소프트웨어기술연구소 인터넷서비스 연구부

Design and Implementation of the CIR-Tree Manager on MIDAS-III for Supporting Efficient Content-Based Image Retrieval

Hee Jong Lee*, Seok-II Song*, Seok Hee Lee*, Jae Soo Yoo*,
Khi Hyung Joe*, Hun Soon Lee**, Jang Sun Lee**

*Dept. of Computer & Communication Eng. Chungbuk National University

**Internet Service Department. Computer and Software Technology Labs.
Electronics and Telecommunications Research Institute.

요 약

최근 이미지 데이터에 대한 요구가 폭발적으로 증가됨에 따라 대용량 이미지 데이터에 대한 저장과 검색에 관한 연구가 활발히 진행되고 있다. 그러나 이미지 데이터는 기존의 텍스트 데이터에 비해 대용량이라는 특성과 비정형인 특성을 가지고 있어 신속하고 효율적인 검색에 많은 어려움이 있다. 본 논문에서는 기존에 이미지 검색을 위해 제안된 인덱스 구조중 고차원 특성을 효율적으로 수용하고 저장공간의 이용률과 검색성능이 뛰어난 CIR-트리를 국내에서 개발된 상용 데이터베이스 시스템인 바다의 하부 저장구조인 MIDAS 기반에서 구현한다 CIR-트리 관리기를 갖는 MIDAS-III에서 K-NN 질의 및 범위 질의가 처리될 때 순차검색에 비해 약 60~99%정도의 검색성능이 향상되었다.

1. 관련 연구

최근 들어 멀티미디어 기술의 발달이 가속화되면서 멀티미디어 데이터에 대한 이용이 폭발적으로 증가해 왔다. 특히, 이미지 데이터에 대한 이용은 여러 분야에 걸쳐 폭넓게 분포되어 있다. 이들 분야에서 사용되는 이미지 데이터의 양은 이미 사람이 관리하기에는 불가능할 정도로 증가해 버렸고 또 증가하고 있다. 따라서, 사용자들이 대용량의 이미지들중 원하는 이미지를 정확하게 검색할 수 있도록 하는 이미지 검색시스템이 요구된다. 기존 이미지 검색 시스템들은 보통 파일시스템에서 구현되어 있거나, 기존의 DBMS의 위에서 사용자 프로그램 수준에서 구현되어 있어 전체적인 수행성능이 낮을 수 밖에 없다. 그러므로, 효율적인 이미지 검색시스템을 구축하기 위해서는 DBMS의 하부 저장 구조를 내용기반 이미지 검색을 위한 색인구조를 수용할 수 있도록 확장해야 한다.

본 논문에서는 국내 기술로 개발된 멀티미디어 DBMS(이하 바다-III)의 하부 저장구조(이하 MIDAS-III)에 내용기반 이미지 검색을 위한 색인구조를 구현한다. 여기에서 구현하는 고차원 색인 구조는 CIR-트리이다

본 논문의 구성은 다음과 같다. 2 장은 관련연구로서 CIR-트리의 특징과 MIDAS-III의 전체적인 구성에 대해 기술하고, 3 장에서는 MIDAS-III 기반에서 CIR-트리 관리기를 설계하고 구현한 내용에 대해서 설명한다. 4 장에서는 성능 평가를 통해 본 논문에서 구현한 CIR-트리 관리기의 타당성을 검증하며 5 장에서 결론을 맺는다.

2. 관련 연구

CIR(Content based Image Retrieval)-트리는 내용기반 이미지 검색을 지원하기 위해 제안된 고차원 색인 구조이다. 이 색인 구조는 기존의 R-트리 계열의 고차원 색인 구조들의 문제점인 차원이 증가

함에 따라 급격히 색인 구조의 성능이 떨어지는 문제점을 개선하고 있다. CIR-트리의 특징을 크게 3가지로 정리할 수 있다. 첫째, 단 말노드뿐 제외한 모든 노드들이 분별력이 있는 특징만을 선정하여 노드를 구성함으로써 효율적으로 고차원의 이미지 특징들을 수용한다. 둘째, 검색영역을 최소화하기 위한 노드의 분할 방법을 제공하며 슈퍼노드 개념을 도입했다. 셋째, 이미지 특징들의 군집화를 위해 부계 중심을 이용하는 보다 개선된 재삽입을 수행한다. 그림 1은 CIR-트리의 전체적인 구조이다.

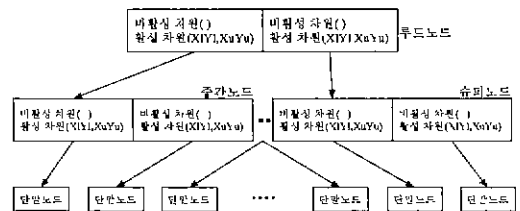


그림 1 CIR-트리의 구조

일반적으로 DBMS[9]는 크게 두 부분으로 나눌 수 있다. 사용자 인터페이스라고 할 수 있는 데이터베이스 언어 처리기 부분과 데이터를 저장하고 하고 트랜잭션을 관리하는 저장시스템이다. 이 중 데이터의 저장 및 트랜잭션관리의 역할을 하는 것이 바로 MIDAS-III이다. 앞서 언급 한대로 MIDAS-III는 바다-III의 하부 저장 시스템으로서 데이터의 저장, 접근, 동시성 및 회복 등을 지원한다. 그림 2에서 MIDAS-III의 전체적인 구성을 보여주고 있다. 데이터를 처리

하는 부분은 디스크에 대한 데이터의 입출력과 메모리 관리 그리고 인덱스 관리, 커서관리, IR-Index 관리 등이다. 트랜잭션을 관리하는 부분은 잠금 관리, 작업관리, 회복관리 부분이다.

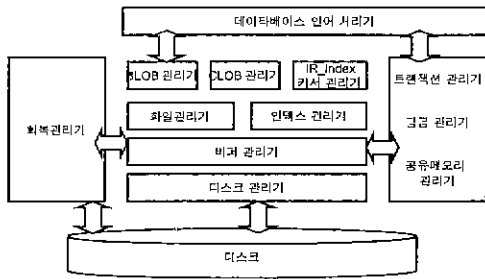


그림 2 MiDAS-III 시스템 내부 구성도

3. CIR-트리 관리기의 설계 및 구현

3.1. CIR-트리 관리기의 설계

실제적인 CIR-트리 관리기의 구성에 대해 설명하기 전에 먼저 CIR-트리 관리기가 MIDAS 내에서 어떻게 위치하는가에 대해서 설명한다. MIDAS 시스템은 데이터의 입력 및 검색을 위해 커서를 사용한다. 이 커서를 통해서 인덱스 나 데이터 파일을 접근할 수 있도록 하고 있다. 하지만 IR-인덱스 관리기는 질의와 부합하는 문서를 계산하는 단계를 거쳐야 하므로 별도의 커서를 두고 있으며 커서관리기와 동등한 위치에 존재한다. 그림 3은 MIDAS에서 CIR-트리 관리기의 위치에 대해서 보여주고 있다. CIR-트리 관리기는 파일 관리기를 통해 데이터 파일에 접근하고 특징벡터, 비퍼 관리기와 디스크 관리기들과 상호 작용하며 일을 수행한다.

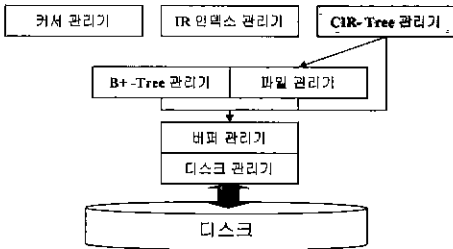


그림 3 MiDAS에서의 CIR-트리 관리기의 위치

그림 4에서 본 논문에서 구현한 CIR-트리 관리기의 구성도를 보여 주고 있다. 이미지 데이터의 특징 벡터는 이미지 검색 엔진에서 이미지 진처리 과정을 거쳐 추출되고 이것은 다시 데이터 파일 내 레코드의 한 필드에 저장한다. CIR-트리 인덱스를 구축하기 위한 입력 데이터는 이미지 데이터와 특징벡터가 저장되어 있는 데이터 파일의 레코드에 대한 식별자가 된다. 이 레코드 식별자와 함께 레코드 내의 특징벡터 필드를 설명해 주는 필드 설명자가 주어지면 데이터 파일로부터 특징벡터를 추출해내고 이 특징 벡터가 레코드 식별자와 함께 CIR-트리 인덱스의 리프 페이지 엔트리로 구성되고 이를 트리에 삽입하게 된다. 구현한 CIR-트리에서는 특징벡터를 적절한 정수로 변환하여 사용한다. 특징벡터가 분포할 수 있는 범위와 이미지 데이터의 개수등을 적절히 고려하여 특징벡터를 일정 범위의 정수로 변환하게 된다. 실제 CIR-트리에 삽입되는 것은 이 정수형태로 변환된 특징벡터이다. 이미지 검색시에도 이와 같은 방법으로 질의 특징벡터를 정수로 변환하여 검색을 수행한다.

구축된 CIR-트리 인덱스들은 각각 그 인덱스에 대한 인덱스 설명자를 갖는다. 이 설명자에는 CIR-트리 인덱스 구축에 사용된 특징 벡터의 차원수, 트리의 깊이, 필드 설명자 등의 정보들이 포함된다. 인덱스 설명자는 MIDAS의 인덱스 카탈로그 파일에 저장되게 되는데 이때 이 인덱스 설명자 레코드 식별자가 구축된 CIR-트리 인덱스를 유일하게 구분해주는 인덱스 식별자가 된다. 그림 5가 이를 설명한다.

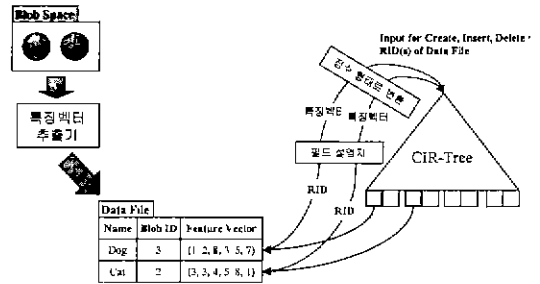


그림 4 CIR-트리 관리기의 구성도-1

CIR-트리 인덱스의 개방은 이 인덱스 식별자로부터 시작한다. 엔트리의 삽입, 삭제 또는 탐색시에는 먼저 이 인덱스 식별자를 통해 인덱스 카탈로그 파일로부터 인덱스 설명자를 읽어 오고 인덱스 설명자를 바탕으로 커서를 생성하게 된다. 이 커서에는 CIR-트리 인덱스의 루트 페이지, 결과 집합, 필드 설명자등을 유지하며 이런 정보들은 실제로 탐색을 하거나 새로운 엔트리를 삽입하거나 또는 엔트리를 CIR-트리 인덱스에서 삭제 할 때 사용된다.

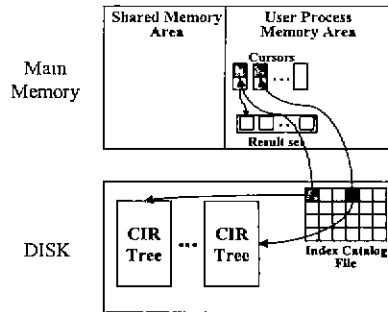


그림 5 CIR-트리 관리기의 구성도-2

3.2. CIR-트리 관리기의 외부 인터페이스

다음은 구현한 CIR-트리 관리기의 외부 인터페이스이다

```

midas_cir_createindex(BaseFileID, FieldDesc, TotalDim,
    Transform(), ActiveDim, CIRIndexID)
midas_cir_destroyindex(CIRIndexID)
midas_cir_opencursor(CIRIndexID, Mode)
midas_cir_closecursor(CIRcsid)
midas_cir_addcursor(CIRcsid, Rid)
midas_cir_addbulk(CIRcsid, Rid, RidFileID, No_of_Rids)
midas_cir_deletecursor(CIRcsid, Rid)
midas_cir_searchimage(CIRcsid, Key_for_PointData, No_of_Rid)
midas_cir_getimage(CIRcsid, Result, No_of_Rid, Navigation)
    
```

구현한 CIR-트리 관리기에서는 탐색이 두 단계로 이루어진다. 먼저 `midas_cir_searchimage`를 통해서 질의에 부합하는 특징벡터들을 갖고 `midas_cir_getimage`를 통해서 원하는 이미지를 기리슨 또는 RID 순으로 읽어 온다.

4. 성능 평가

본 논문에서는 순차 탐색과 비교를 통해 구현한 CIR-트리 관리기의 타당성을 증명한다. 구현에 사용된 시스템은 Solars 2.6.x 운영 체제에 Sun UltraSPARC-II, 366MHz CPU 2 개를 장착하고 있으며 사용된 컴파일러는 gcc 2.7.1 이다. 실험에 사용한 데이터 집합은 전체 10차원에 합성차원 5이며 5만개 데이터를 사용하였다.

CIR-트리 관리기의 타당성을 보이기 위해 CIR-트리를 이용해 K-

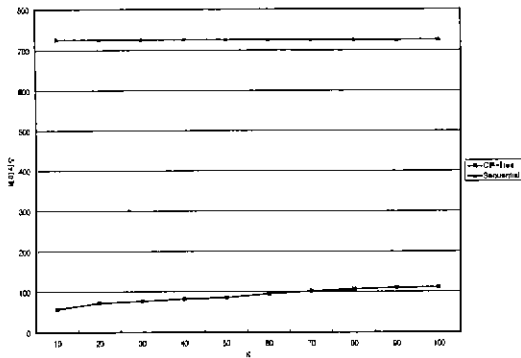


그림 6 K-최근접 질의 (페이지수)

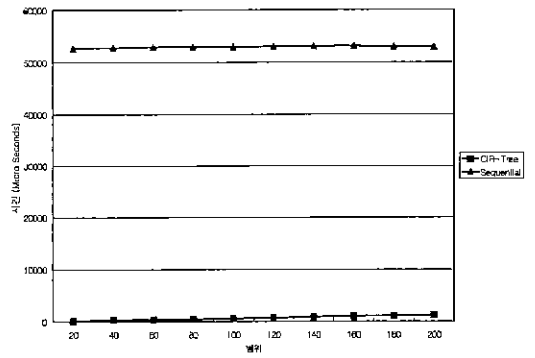


그림 9 범위 질의 (시간)

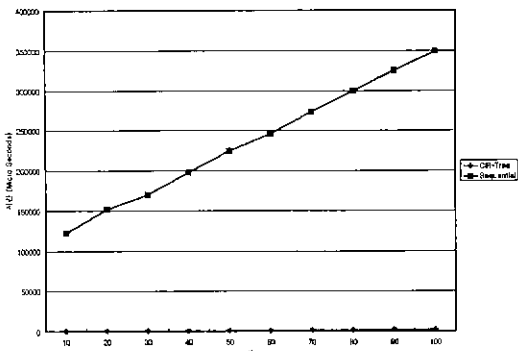


그림 7 K-최근접 질의 (시간)

최근접 질의와 범위 질의를 수행할 때 접근한 페이지의 수와 검색시간 그리고 순차 탐색을 이용해 K-최근접 질의와 범위 질의를 수행할 때 접근한 페이지수와 시간을 각각 비교한다.

그림 6은 K-최근접 질의를 각각 CIR-트리를 통해서 수행 했을 때와 순차 탐색을 이용하여 수행했을 때 접근한 페이지의 수를 비교한 것이다. CIR-트리를 통해 수행했을 때가 그렇지 않을 때 보다 약

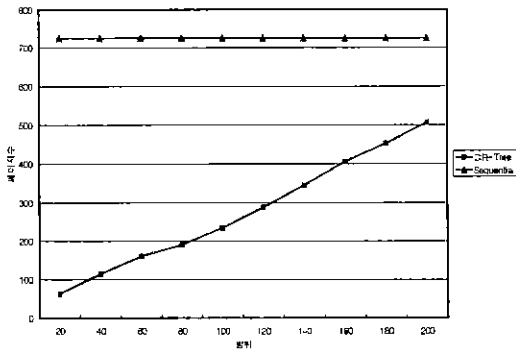


그림 8 범위 질의 (페이지수)

87 % 정도의 성능 향상을 얻을 수 있었다. 그림 7은 같은 상황에서 시간을 가지고 비교한 것이다. 이 경우에는 약 99 % 정도의 성능 향상을 얻는 것을 볼 수 있었다.

그림 8과 9는 각각 범위 질의를 수행했을 때 그 접근한 페이지수와 수행시간을 가지고 비교한 것이다. 이 경우에도 각각 60% 와

98%의 성능 개선 효과를 볼 수 있었다. 평균적으로 보면 페이지 접근 회수 보다 수행 시간 측면에서 더 높은 성능 개선 효과를 볼 수 있었는데 그 이유는 순차 탐색의 경우에는 접근하는 모든 페이지는 모두 처음 접근하는 페이지이지만 CIR-트리의 경우에는 인터넷 페이지에 해당하는 페이지들은 이미 접근했던 페이지일 확률이 있기 때문이다.

5. 결론

본 논문에서는 보다 효율적인 내용기반 이미지 검색을 지원하기 위해 MIDAS-III에서 CIR-트리 관리기를 구현하였다. 구현한 CIR-트리 관리기는 순차 검색과 비교했을 때 매우 뛰어난 성능을 보였다. 이것은 CIR-트리 관리기가 MIDAS에 포함됨으로써 그 이전보다 훨씬 더 효율적인 내용기반 이미지 검색을 지원할 수 있다는 것을 의미한다.

향후 연구에서는 삽입연산 수행시 재삽입을 수행할 때가 수행하지 않을 때에 비해 수행 시간이 상대적으로 매우 긴 단점이 있는데 이를 보완한다. 그리고, 본 논문에서 구현한 CIR-트리 관리기는 동시성 및 회복이 고려되지 않았는데 향후 연구에서는 회복 및 동시성을 고려한 CIR-트리 관리기를 MIDAS에서 구현 할 것이다.

참고 문헌

- [1] 이석희, 최길성, 유재수, 조기형, "CIR-트리 : 내용기반 이미지 검색을 위한 효율적인 고차원 색인기법", 한국정보과학회 97 가을 학술발표(I), pp.349-352, 1997.
- [2] 이석희, 송석일, 유재수, "내용기반 이미지 검색을 위한 고차원 색인구조", 한국정보과학회 데이터베이스연구회 논문지, 제14권, 제 4호, pp.53-68, 1998
- [3] "마이더스 인터페이스 규격서", 전자통신연구소
- [4] K.I. Lin, H. Jagadish, and C Faloutsos, "The TV-tree -- An Index Structure for High Dimensional Data," VLDB Journal, Vol 3, pp517-542, 1994
- [5] S. Berchtold, D. A Keim, H-P. Kriegel, "The X-tree : An Index Structure for High-Dimensional Data", International Proceedings of the 22nd VLDB Conference
- [6] D. A White and R Jam, "Similarity Indexing : Algorithm and Performance", International Proceedings of The SPIE Storage and Retrieval for Image and Video Database IV, Vol 2670, pp.62-75, 1996.
- [7] N. Beckmann, H.P. Kriegel, R. Schneider and B Seeger "The R-tree : An Efficient and Robust Access Method for Points and Rectangles," ACM SIGMOD, pp.322-331 1990.
- [8] Roussopoulos N, Kelley S, Vinvent F., "Nearest Neighbor Queries," ACM SIGMOD pp.71-79. 1995.
- [9] 박치환 외 13 인, "바다 DBMS를 중심으로 한 데이터베이스 관리 시스템 구조", 한국전자통신연구원, 1997