JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 18, No. 1 : 103~108 Jan. 2014

플래쉬 메모리기반 저장장치에서의 공간분할기법 색인의 성능 평가

김동현*

The Performance Evaluation of a Space-Division typed Index on the Flash Memory based Storage

Dong Hyun Kim*

Division of Computer Information Engineering, Dongseo University, Busan, 617-715, Korea

요 약

스마트폰과 같은 휴대용 기기에서 많이 사용되는 플래쉬 메모리는 비휘발성 저장장치로 작은 크기에 대용량 데이터를 안정적으로 저장할 수 있는 장점을 가지고 있다. 플래쉬 메모리에 저장된 대용량 데이터에 대한 질의 연산을 효율적으로 처리하기 위하여 색인을 사용해야 한다. 그러나 플래쉬 메모리는 쓰기 연산의 속도가 느리고 덮어쓰기 연산을 지원하지 않기 때문에 기존의 색인을 평가하고 개선점을 파악할 필요가 있다. 이 논문에서는 플래쉬 메모리에 적용한 공간분할 기법의 공간 색인에 대한 성능을 평가한다. 이를 위하여 고정그리드파일을 구현하여 다양한 환경에서 질의 연산과 변경 연산의 평균 연산 수행 속도를 측정한다. 그리고 자기디스크 저장장치에서의 수행속도와 비교한다.

ABSTRACT

The flash memory which is exploited on hand-held devices such as smart phones is a non-volatile storage and has the benefit that it can store mass data at a small sized chip. To process queries on the mass data stored in the flash memory, the index scheme should be exploited. However, since the write operation of the flash memory is slower than the read operation and the overwrite is not supported, it is required to reevaluate the performance of the index and find out the drawbacks. In this paper, we evaluate the performance of a space division typed index scheme on the flash memory. To do this, we implement the fixed grid file and measure the average speeds of the query and update processing on a various condition and compare the value of the flash memory with that of the magnetic disk.

키워드 : 플래쉬 메모리, 공간분할기법 색인, 색인, 성능평가, 고정그리드파일

Key word : flash memory, space division typed index, index, performance evaluation, fixed grid file

접수일자 : 2013. 09. 12 심사완료일자 : 2013. 10. 17 게재확정일자 : 2013. 10. 31

*** Corresponding Author** Dong Hyun Kim(E-mail:pusrover@dongseo.ac.kr , Tel:+82-51-320-1801)
Division of Computer Information Engineering, Dongseo University, Busan, 617-715, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2014.18.1.103 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

플래쉬 메모리는 비휘발성 메모리로써 적은 전력을 소모하고 휴대성이 좋은 특성을 가지고 있다. 따라서 PDA, 스마트폰과 같은 휴대용 기기의 데이터 저장 장치로 주로 사용되었으며 최근에는 메모리칩의 가격이 저렴해짐에 따라 기존의 자기디스크 저장장치를 대신하여 SSD(Solid State Drive) 같이 보조 저장장치로도 사용되고 있다.

NAND-플래쉬 메모리는 32섹터로 구성된 메모리 블럭으로 구성되고 각 섹터는 512바이트 크기의 데이터 영역과 16바이트의 부가 영역으로 구성된다[2-4]. 플래쉬 메모리는 데이터에 접근하기 위하여 전기신호를 사용하기 때문에 자기디스크 저장장치와 비교하여 저장 위치와 상관없이 균일한 접근 속도를 제공한다. 그러나 쓰기 연산의 속도가 읽기 연산보다 매우 느리고 특히 덮어쓰기 연산을 지원하지 않기 때문에 동일 위치에 데이터를 저장하기 위하여 지움연산을 먼저 수행해야 하는 특징을 가진다.

저장할 데이터의 양이 적은 휴대용기기에서 플래쉬 메모리가 사용되는 경우에 부가적인 구조인 색인을 사용할 필요가 적다. 그러나 워크스테이션 또는 서버의 보조 저장장치로서 대용량 데이터를 저장할 경우에 효율적인 검색을 위하여 색인 구조를 사용해야 한다. 그러나 기존의 색인 성능 평가는 자기디스크 저장장치에서 수행되었으며 플래쉬 메모리는 자기디스크와는 다른 특성을 가지고 있기 때문에 플래쉬 메모리 상에서의 색인의 성능을 재평가하여 색인의 개선점을 파악할 필요가 있다.

본 논문에서는 플래쉬 메모리상에서 공간분할 기법의 공간 색인의 성능을 평가한다. 이를 위하여 공간분할 기법의 대표적인 고정그리드파일 색인을 구현한다. 그리고 스케일 크기, 버켓의 크기, 데이터 집합의 크기에 따른 평균반응시간을 자기디스크에서의 평균반응시간과 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련된 연구 동향에 대하여 기술하며 3장에서는 실험 환경과 실험 결과에 대하여 기술한다. 마지막으로 4장에서는 결론 및 향후 연구를 기술한다.

Ⅱ. 관련 연구

그리드 화일[1]은 해쉬 기반의 색인으로 공간 분할 기법의 대표적인 색인이다. 일반 해쉬 색인과는 달리 다차원 데이터 공간을 분할하는 색인으로 그리드 디렉토리와 데이터 버켓으로 구성되고 그리드 디렉토리는 스케일에 따라 각 데이터 차원을 분할한다. 그리고 공간 데이터는 각 그리드 디렉토리의 셀에 연결된 데이터 버켓에 저장된다. 저장된 데이터에 접근하기 위해서는 공간 좌표값을 해쉬하여 계산한 그리드 디렉토리의 셀에 먼저 접근한 다음에 셀과 연결된 데이터 버켓에 접근한다. 그리드 화일은 점형의 공간 데이터는 효율적이지만 선 또는 면형의 공간 데이터는 비효율적인 단점이 있다.

BFTL(B-Tree Flash Translation Layer)[2]는 플래쉬 변환층(FTL, Flash Translation Layer)에서 B-Tree를 구현한 기법으로 유보버퍼, 노드변환테이블, 그리고 완료정책으로 구성된다. 유보버퍼는 새로운 키값이 삽입될 때 구성된 단말노드의 색인유닛을 임시로 저장하는 공간이다. 유보버퍼에 여유 공간이 없을 경우 완료정책에 따라 유보버퍼의 색인유닛이 플래쉬 메모리에 저장된다. 이때 서로 다른 노드의 색인유닛이 동일섹터에 저장될 수 있으므로 각 노드의 색인유닛 저장주소를 노드변환테이블에서 관리한다.

BOF(B-Tree On Flash memory)[3]는 BFTL기법의 단점을 개선한 기법으로 유보버퍼내에 있는 동일노드의 색인유닛들을 플래시메모리의 동일 섹터에 저장한다. 따라서 한 섹터에 하나의 색인 정보만을 저장하기 때문에 한 번의 읽기 연산을 통하여 한 노드의 정보를 읽을 수 있고 노드변환테이블을 사용하는 부하가 없다. 단, 새로운 키값을 삽입할 때 먼저 삽입될 단말 노드를 검색한 후에 해당 노드의 데이터와 유보버퍼의 색인유닛과의 노드 병합을 먼저 실시한다. 그리고 병합된 노드에 새로운 키값을 삽입한다.

[4]에서는 R-Tree를 플래쉬 메모리에서 사용하기 위한 노드압축기법을 제안하였다. 데이터에 대한 삽입 또는 삭제가 발생하면 이를 위한 유보버퍼의 객체를 생성한다. 유보버퍼 객체는 최소경계사각형, 연산종류 그리고 데이터에 대한 포인터로 구성된다. 유보버퍼에 여유 공간이 없으면 유보버퍼의 객체를 색인유닛으로 변경하고 플래쉬 메모리의 페이지에 압축하여 저장한다. 이때 동일노드의 색인유닛이 동일 페이지에 저장되지 않

기 때문에 노드변환테이블을 사용한다.

[5]에서는 MR-Tree를 플래쉬 메모리에서 사용하기 위한 기법을 제안하였다. 쓰기 연산의 횟수를 줄이기 위하여 노드의 삽입/삭제 연산의 크기가 플래쉬 메모리의 페이지 크기가 될 때까지 버퍼에 저장한다. 누적된 연산들의 크기가 페이지 크기와 동일해 질 때 비로소 플래쉬 메모리의 페이지에 적용한다. 이를 위하여 플래쉬 메모리의 페이지와 동일한 크기의 섹터버퍼를 사용한다.

플래쉬 메모리 기반의 저장장치에서 색인과 관련한 기존의 연구는 데이터 유도형 색인에서 많이 사용되는 트리 구조의 색인에 대하여 연구되어졌다. 그러나 다차원 공간 색인 구조는 데이터 유도형 색인 외에 공간분할형 색인도 다수 사용되기 때문에 공간분할 색인에 대한 성능평가가 필요하다.

Ⅲ. 관련 연구

3.1. 그림 편집

색인의 성능을 평가하기 위한 측정값은 크게 두 가지가 있다 첫 번째는 디스크 접근횟수이다. 일반적으로 디스크 색인의 성능을 평가하기 위한 방법으로 디스크 접근 횟수와 평균 디스크 접근 시간을 이용하면 균일한 측정값을 계산할 수 있다. 두 번째는 연산의 수행 횟수이다. 메인메모리 색인의 성능을 평가하기 위한 방법으로 중앙연산장치의 연산시간을 이용하여 측정값을 계산할 수 있다. 그러나 플래쉬 메모리에서의 색인 성능과 자기디스크에서의 성능을 비교하기 위하여 본 논문에서는 데이터집합에 대한 모든 연산을 수행하기 위해 소요되는 평균수행시간을 측정한다.

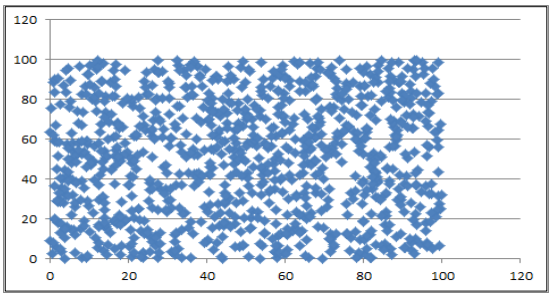

그림 1. 실험 데이터 분포도
Fig. 1 Data Distribution for evaluation

실험을 위한 공간 데이터는 고정그리드파일에 적합한 점데이터를 사용한다. 비교 평가하기 위한 데이터 집합의 크기는 3가지 종류를 사용하고 데이터 분포 유형은 그림 1과 같이 균등 분포형을 사용하고 범위는 0.0 ~ 100.0로 일반화한다.

기존 그리드파일의 성능은 그리드 디렉토리의 스케일 수와 데이터 버켓의 크기에 따라 좌우된다. 스케일 수가 클수록 버켓의 오버플로우 횟수는 줄어들지만 질의처리시 접근해야 하는 그리드 디렉토리 셀의 수는 늘어난다. 또한 버켓에 저장되는 점 데이터의 수가 많을수록 오버플로우 횟수와 검색하기 위한 오버플로우 버켓의 수는 줄어들지만 점 데이터를 검색하기 위하여 한 번에 접근하는 버켓의 크기가 증가한다. 본 논문에서는 스케일과 버켓의 크기를 5가지 종류로 나누어 실험한다. 표 1은 실험조건들을 보여준다.

표 1. 실험 조건
Table. 1 Evaluation Condition

구분	종류
실험저장장치	자기디스크, SD 메모리(Flash), USB 메모리(Flash)
데이터집합 크기	점데이터 1,000개 , 5,000개, 10,000개
스케일 수	4X4, 32X32, 64X64, 256X256, 512X512
버켓 크기	1개, 2개, 5개, 10개, 21개

실험을 위한 저장장치는 자기디스크, SD메모리, USB메모리인 3가지 종류를 사용한다. 각각 4k바이트 크기의 데이터에 대하여 0.342/0.605, 3.265/0.051, 3.544/0.062(Mbte/sec)의 읽기/쓰기 성능이다.

3.2. 실험 결과

그림 2는 세 종류의 저장장치에 대하여 4x4스케일 색인을 구축할 때 색인 구축 시간을 비교한 그래프이다. (a)는 버켓의 크기가 1인 경우이고 (b)는 21인 경우이다. 가로축은 데이터집합의 크기이며 세로축은 구축에 필요한 평균수행시간이다. 그림 2에서 보듯이 자기디스크 저장장치와 비교하여 SD메모리는 약 4% ~ 6%의 성능만을 보여주고 있으며 USB메모리는 17% ~ 20%의 성능을 보여주고 있다. 스케일의 수를 512x512로 확장하여 실험하여도 그림 3에서 보듯이 유사하게 각각 2% ~

3%, 16% ~ 20%의 결과를 보여준다. 이는 플래쉬 메모리가 자기디스크에 비하여 쓰기연산 속도가 매우 느리기 때문이며 특히 덮어쓰기 작업을 수행하기 위하여 지움연산을 추가로 수행하는 비용이 매우 많다는 것을 보여준다.

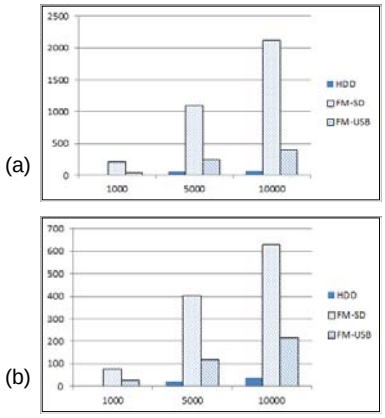

그림 2. 4x4스케일 색인 구축 시간
Fig. 2 Index Building Time on 4x4 scale

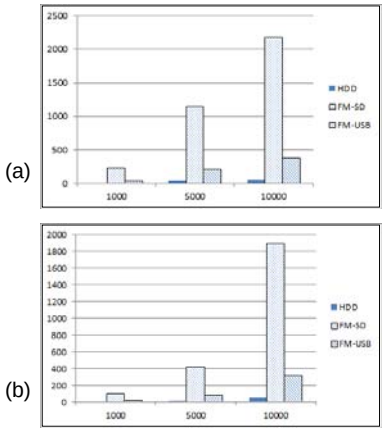

그림 3. 512x512스케일 색인 구축 시간
Fig. 3 Index Building Time on 512x512 scale

그림 4는 질의 연산과 변경 연산의 비율을 1:9로 변경 연산의 비율이 매우 높은 환경에서 수행 시간을 비교한 그래프이다. (a)는 버켓의 크기가 1인 경우이고 (b)는 21인 경우이다. (a) 그래프에서 보듯이 버켓의 크기가 1인 경우에는 SD는 약 5% ~ 7%의 성능만을 보여주고 USB의 경우에는 8 ~ 12%의 성능을 보여준다. 그러나 (b) 그래프에서 보듯이 버켓의 크기가 21로 페이지의 크기와 동일한 경우에는 각각 6% ~ 8%, 36% ~ 50%의 성능을 보여준다. 이는 버켓의 크기가 플래쉬 메모리의 페이지 크기와 유사해질 경우 쓰기 연산의 성능이 개선되기 때문이다.

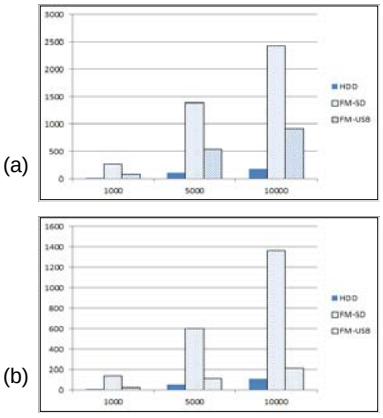

그림 4. 질의:변경 = 1:9
Fig. 4 Query:Modification = 1:9

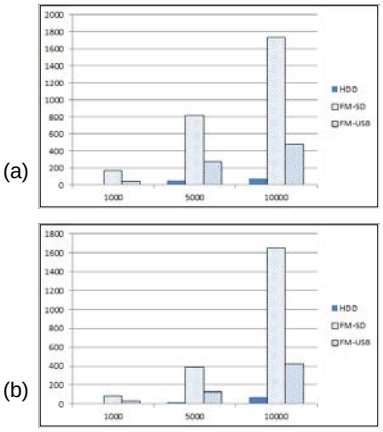

그림 5. 질의:변경 = 5:5
Fig. 5 Query:Modification = 5:5

그림 5는 질의와 변경 연산의 비율이 동일하게 수행하였을 때의 실험 결과이고, 그림 6은 질의 연산과 변경 연산의 비율이 9:1로 질의 연산을 많이 수행하였을 때의 결과이다. 각각 (a)는 버켓의 크기가 1인 경우이고

(b)는 크기가 21인 경우이다. 각 그림에서 보듯이 동등한 비율로 수행되거나 또는 질의 연산을 많이 수행하더라도 SD는 4% ~ 7% 그리고 USB는 11% ~ 18%의 성능만을 보여주고 있다. 이는 질의 연산의 비율이 커지더라도 변경 연산비용 때문에 자기디스크와 비교하여 색인의 성능이 저하되는 것을 보여준다.

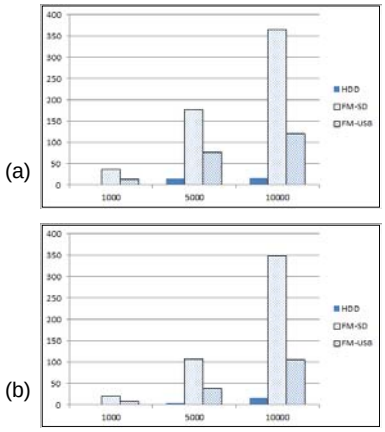

그림 6. 질의:변경 = 9:1
Fig. 6 Query:Modification = 9:1

플래쉬 메모리라고 할지라도 종류에 따라 질의 연산에 소요되는 비용이 서로 다르다는 것은 그림 7의 그래프에서도 확인할 수 있다. 그림 7은 512x512 스케일, 버켓 크기 21인 상태에서 질의 연산을 수행하였을 때 수행시간을 보여준다. 자기디스크와 비교하여 각각 5% ~ 8%, 1% ~ 3%의 성능만을 보여준다. 이는 플래쉬 메모리에서 임의접근 방식 읽기 연산의 속도가 매우 느리며 색인이 사용되는 플래쉬 메모리의 종류에 따라서도 매우 다름을 보여준다.

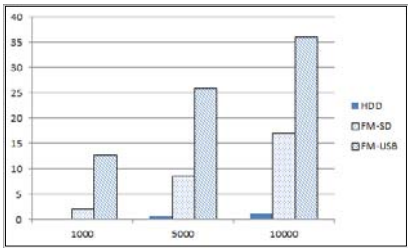

그림 7. 질의 연산 시간
Fig. 7 Query Processing Time

저장 장치의 특성 외에 고정그리드 색인의 성능에 영향을 줄 수 있는 요소는 스케일의 수와 버켓의 크기이다. 그림 8은 버켓의 크기 1인 상태에서 스케일의 수를 변화시켰을 때의 색인 구축 시간이다. 그림에서 보듯이 자기디스크는 큰 변화가 없으나 플래쉬 메모리는 SD나 USB도 동일하게 스케일이 32 그리고 64인 경우 구축시간이 적게 소요됨을 알 수 있다.

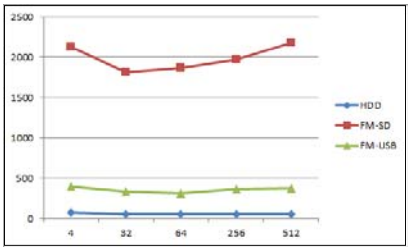

그림 8. 스케일에 따른 색인구축시간
Fig. 8 Index Building Time by Scale

그림 9는 4x4 스케일 상태에서 버켓 크기의 변화에 따른 처리 시간을 보여준다. (a)는 색인을 구축할 때 소요 시간이며 (b)는 색인을 변경할 때 소요시간이다. 그림 9에서 보듯이 버켓의 크기가 21인 경우처럼 쓰기 연산의 크기가 페이지의 크기와 유사해짐에 따라 색인의 성능이 개선되는 것을 보여준다. 이러한 현상은 (b)와 같이 색인 변경 연산의 횟수가 많은 경우에도 동일하다.

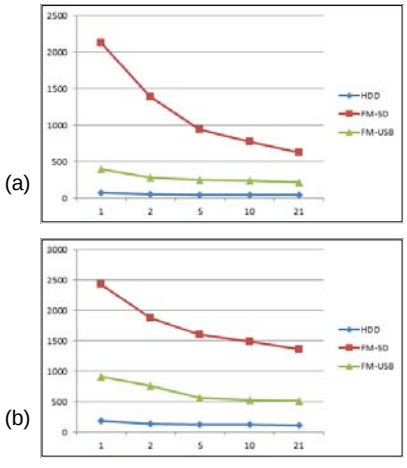

그림 9. 버켓의 크기에 따른 소요시간
Fig. 9 Processing Time by Bucket Sizes

실험 결과 그래프에서 보듯이 플래쉬 메모리에서 색인을 구축하여 운용할 때 쓰기 연산의 비용이 크기 때문에 질의 연산의 비율이 높더라도 자기디스크보다 색인의 성능이 매우 떨어지는 것을 볼 수 있다. 특히 덮어쓰기의 횟수가 많아질수록 색인의 성능은 매우 저하된다. 또한 색인의 구조상 색인 연산을 위하여 임의접근 방식을 통한 데이터 접근 횟수가 매우 많다. 그러한 경우에 플래쉬 메모리 저장장치가 자기디스크에 비하여 성능이 매우 떨어지며 이러한 현상은 플래쉬 메모리의 종류에 따라 매우 큰 차이를 보여준다. 하지만 버켓의 크기가 플래쉬 메모리의 페이지 크기와 근접해질수록 색인의 구축 시간이나 색인의 변경연산 소요 시간이 매우 개선되는 것을 볼 수 있다. 따라서 고정그리드 색인을 개선하기 위해서 연산의 대상이 되는 버켓의 크기를 페이지의 크기로 설정하여 주고 특히 쓰기 연산과 덮어쓰기 연산의 횟수를 줄여야 한다.

Ⅳ. 결론 및 향후 연구

플래쉬 메모리는 비휘발성 저장장치로 작은 크기에 많은 데이터를 안정적으로 저장할 수 있기 때문에 스마트폰, USB와 같은 휴대용 기기에서 주로 사용되어 진다. 플래쉬 메모리의 저장용량이 커지고 워크스테이션과 같은 비휴대용 기기의 보조저장장치로 사용됨에 따라 대용량 데이터를 효율적으로 처리하기 위한 색인을 사용해야 한다. 그러나 플래쉬 메모리의 특성 때문에 기존의 색인을 사용하면 성능이 저하될 수 있다.
이 논문은 플래쉬 메모리에 데이터 공간분할 기법인 다차원 공간 색인의 성능을 평가한다. 이를 위하여 고정그리드 화일 색인을 플래쉬 메모리에서 구현하고 다양한 조건에서 성능 평가를 수행한다. 실험 결과를 살펴보면 쓰기와 덮어쓰기 연산의 비용이 매우 높으며 질의를 수행하기 위해 색인에 대하여 임의 접근 방식을 사용하면 색인의 성능이 매우 저하됨을 알 수 있다. 본 논문의 결과는 산업적으로 플래쉬 메모리에서의 색인 구현시 활용될 수 있으며, 향후 연구로는 고정그리드 화일 색인에서 쓰기와 덮어쓰기 연산의 횟수와 질의 처리를 위한 데이터 임의접근 횟수를 줄이기 위한 기법에 대한 연구가 필요하다.

감사의 글

이 논문은 2012년도 동서대학교 “Dongseo Frontier Project” 지원에 의해 이루어진 것임.

REFERENCES

[1] J.Nievergelt et al.,“The Grid File:An Adaptale, Symmetric Multikey File Structure”, *Trends in Information Processing*, LNCS 123, pp.236-251, 1981.

[2] Chil-Hsien Wu et al., “An Efficient B-Tree Layer for Flash Memory Storage System”, *Real-Time and Embedded Computing Systems and Applications*, LNCS 2968, Springer, pp.409-430, 2003.

[3] Junghyun Nam et al., “The Efficient Design and Implementation of the B-Tree on Flash Memory”, *Proc. of the 25th KISS Fall*, pp.55-57, 2005.

[4] Chin-Hsien Wu et al., "An Efficient R-Tree Implementation over Flash-Memory Storage Systems", *ACM Intl. Symp. on Advances in Geographic Information Systems*, ACM, pp.17-24, 2003.

[5] Hyun Seung Lee et al., "Performance of Index trees on Flash Memory", *Intl. MultiConf. on Computer Science and Information Technology*, pp.725-734, 2007.

김동현(Dong Hyun Kim)

부산대학교 컴퓨터공학과 공학박사
※관심분야: 공간데이터베이스, LBS, 센서데이터베이스, 빅데이터