

# 멀티 플랫폼 환경에서 효율적인 공간 연산을 위한 객체의 근사 표현 기법

강구안 · 김진덕  
동의대학교

## Approximation Methods for Efficient Spatial Operations in Multiplatform Environments

Gu-an Kang · Jin-Deog Kim  
Donggeui University  
E-mail : jdk@donggeui.ac.kr

### 요 약

빠른 질의 처리를 요구하는 공간 데이터베이스 시스템은 주로 최소경계사각형을 이용한 여과단계를 수행한 후 후보 객체에 대해 정제 단계를 수행한다. 대부분의 공간 연산은 빠른 여과 단계를 요구하지만, 처리 능력이 떨어지는 단말기에서는 여과율을 높여 정제 단계의 수행 횟수를 줄이는 것이 필수적이다. 그리고 메모리 용량이 부족한 모바일 단말기는 최소경계사각형을 간략히 표현해야 한다.

이 논문에서는 멀티 플랫폼 환경에서 공간 연산의 효율적인 여과 단계를 위한 공간 객체의 다양한 근사 표현 방법을 제안한다. 또한 2차원 공간 색인의 80% 이상을 차지하는 최소 경계 사각형 정보를 압축하기 위한 방법을 제안한다. 그리고 표현 용량과 여과 효율 및 여과 처리 속도 측면에서 각 방법의 장단점을 분석한다.

### ABSTRACT

Spatial database systems achieve filtering steps with MBR(Minimum Bounding Rectangle) for efficient query processing, and then carry out refinement steps for candidate objects. While most operations require fast execution of filtering, it is necessary to increase the filtering rates and reduce the number of refinement steps in the low computing powered devices. The compact representation method is also needed in the mobile devices with low storage capacity.

The paper proposes various approximation methods for efficient spatial operations in the multiplatform environments. This paper also designs a compression technique for MBR, which occupies almost 80% of index data in the two dimensional case. We also analyze the advantages and drawbacks of each method in terms of space utilization, filtering efficiency and speed.

### 키워드

근사 표현, 최소경계 사각형, 공간 연산, 멀티플랫폼

### 1. 서 론

최근 공간 데이터베이스 시스템을 이용한 응용이 다양화되고, 운용되는 하드웨어 또한 다양해지고 있다. 인터넷 GIS, LBS, PDA와 휴대폰과 같은 모바일 휴대장치[1]를 기반으로 하는 GIS 등등이 대표적인 예이다. 이와 같은 환경에서는 공간 연

산의 여과 단계에서는 저장 장치의 표현 용량, 여과 효율, 여과 처리 속도 측면에서 제각기 다른 특성을 요구한다. 예를 들어 고속의 프로세서 및 병렬 처리 환경에서는 여과 효율보다는 간단하면서도 빠른 여과 단계를 선호하는 반면 저속의 프로세서를 가진 모바일 휴대 장치 환경에서는 비용이 매우 큰 정제 단계의 수행 횟수를 줄이기

위해 저장 용량은 다소 많이 차지하지만 여과 효율이 좋은 여과 단계를 택한다.

그럼에도 불구하고 지금까지의 여과 단계에서의 근사 표현법은 거의 대부분 객체의 최소 경계 사각형을 이용하였다[2]. 그리고 모바일 단말장치는 기존 PC에 비해 작은 용량의 저장 장치와 낮은 성능의 프로세서를 가진다. 반면 공간 데이터는 대용량이며, 공간 연산 비용 또한 매우 크다. 그러므로 지금까지 연구되어온 PC 기반 공간 색인 기법을 소형화해야 함과 동시에 좋은 여과 효율을 갖도록 수정해야 한다.

이 논문에서는 공간 연산의 여과 단계를 위한 공간 객체의 다양한 근사 표현 방법을 제안한다. 구체적으로 기존 MBR의 저장용량을 차지하지만 여과 효율이 적당한 기울어진 사각형을 이용한 근사 표현 방법을 제안하고, 저장용량과 여과 처리 속도 측면에서는 많은 비용이 요구되지만 매우 좋은 여과 효율을 보이는 자유 사각형을 이용한 근사 표현 방법을 새롭게 제시한다. 그리고 공간 효율성을 극대화하기 위해 공간 색인 중 약 80%를 차지하고 있는 MBR을 압축해서 표현하는 기법을 제안하고, 압축으로 인한 여과 효율의 저하가 발생하지 않음을 보이고자 한다. 또한 표현 용량과 여과 효율 및 여과 처리 속도 측면에서 각 근사 표현법의 장단점을 분석한다. 제안한 근사 표현 방법들은 다양한 응용 및 사용환경에 따라 적절히 활용될 수 있다.

이 논문의 구성은 다음과 같다. 제 2장에서는 근사 표현의 정의 및 필요성에 대해 언급하여, 제 3장에서는 각 근사 표현 기법을 설명하고 새로운 MBR 압축 기법을 설명한다. 그리고 각 근사 표현법의 장단점 및 공간 색인에 적용하는 방법을 설명하며, 제 4장에서는 결론 및 향후 연구 과제를 다룬다.

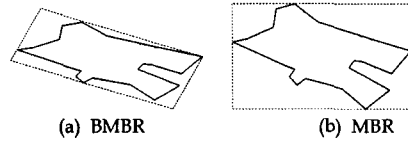
## II. 본 론

지금까지 연구된 공간 색인에서 기본적으로 사용하고 있는 최소경계사각형 이외에 이 논문에서는 기울어진 사각형을 이용한 근사 표현법과 자유 형태의 사각형을 이용한 근사 표현법을 제안한다. 또한 최소경계사각형의 표현 용량을 줄이기 위한 방법을 설계한다.

### 3.1 기울어진 최소경계사각형 근사

이 논문의 기울어진 최소 경계 사각형(Biased MBR)은 다음과 같은 성질을 갖는다. BMBR은 기존의 MBR과 모양은 직사각형으로 같지만 기울어진 각도 값을 갖는다. 그림 1(a)의 점선은 MBR을 나타낸 것이며, 그림 1(b)는 기존의 MBR을 나타낸 것으로서 BMBR에 비해 면적이 클 수 있다. 근사 표현의 면적이 크다는 것은 객체 이외의 지역도 여과 단계의 결과로 출력되므로 여과 효율이 낮아짐을 의미한다. 이와 같은 BMBR의 표현은

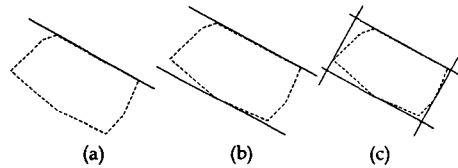
MBR을 회전했을 때 생기는 기존 MBR의 좌하점과 우상점 및 기울어진 각도 값으로 표현한다.



[그림 1] BMBR과 MBR

BMBR의 생성을 위해 두 가지 방법을 제시한다. 첫 번째 방법은 공간 객체의 변을 기준으로 BMBR을 생성하는 것이고, 두 번째 방법은 공간 객체의 중심선을 기준으로 생성하는 방법이다. 첫 번째 방법은 다음과 같은 단계를 거친다.

- 1) 다각형을 블록화시킨다. 다각형의 블록화는 공간 객체의 다각형을 Graham 스캔 알고리즘[3]으로 얻어진 최외곽점을 연결하면 된다.
- 2) 블록화된 다각형에 대해 하나의 변에 대해 접선을 긋고(그림 2(a)), 이 접선에 평행하면서 반대편 꼭지점에 접하는 선을 긋는다(그림 2(b)). 그 뒤 두 직선에 수직이면서 양 끝 꼭지점에 접하는 두 선을 그어 만나는 네 꼭지점을 연결하면 BMBR이 된다(그림 2(c)).



[그림 2] BMBR 생성방법1

단, 단계 2는 각 변을 기준으로 모든 기울어진 사각형을 만든 뒤 가장 작은 면적을 갖는 것을 BMBR로 정하여 최적의 BMBR을 구한다. 그러나 이와 같은 방법은 블록화된 다각형의 복잡도가 높을수록 매우 많은 기울어진 사각형이 존재하므로 BMBR 생성에 많은 시간이 소요된다.

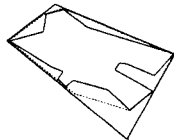
따라서 응용에 따라 매우 높은 여과 효율을 요구하는 경우에는 모든 경우의 수를 고려하며 BMBR을 구하고, 보다 빠른 BMBR을 생성하고자 할 경우에는 가장 긴 변에 접하는 선분을 기준선으로 BMBR을 구한다.

BMBR의 PIP(Point In Polygon) 알고리즘은 주어진 점 X,Y 좌표를 좌하점 기준으로 -8만km 변환한 뒤 기존 MBR에 적용하는 비교 구문을 BMBR에 적용하면 된다. 이는 BMBR을 -8만km 변환하면 MBR과 똑 같은 모양이 되기 때문이다.

### 3.2 자유형의 최소경계사각형 근사

이 논문의 자유형의 최소 경계 사각형(FreeStyle MBR)은 다음과 같은 성질을 갖는다. FSMBR은 다양한 형태의 공간 객체를 둘러싸는 최소의 블록 사각형으로서 공간 객체의 형태에 최대한 근접하면

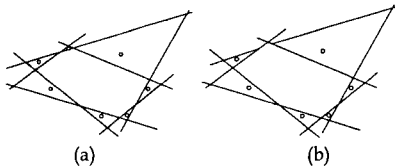
서도 4점으로 표현한다. 그림 3의 안쪽 실선은 실제 공간 객체를 나타낸 것이며, 그림의 점선은 블록화된 다각형이며, 바깥쪽 실선은 이 논문에서 정의한 FSMBR을 나타낸 것이다. FSMBR은 기존의 MBR이나 BMBR에 비해 보다 작은 면적을 차지하여 좋은 여과 효율을 보여주는 장점이 있는 반면 4점으로 표현되기 때문에 저장장치의 표현량이 많아지며, 여과 단계의 PIP 알고리즘 또한 기존 근사 표현법보다 복잡도가 높다는 단점이 있다.



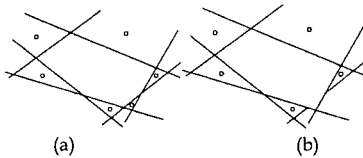
[그림 3] FSMBR

FSMBR을 생성은 다음과 같은 단계를 갖는다.

- 1) 공간객체다각형을 블록화한다(그림 3 점선).
- 2) 블록 다각형의 각각의 면의 연장선을 긋는다. 이 때 각 변마다 그려진 연장선들이 만나는 점을 꼭지점으로 하는 삼각형을 생성된다. 그림 4(a)에서 블록다각형이 7각형이므로 7개의 삼각형이 생성되었다.
- 3) 생성된 삼각형 면적으로 오름차순 정렬한다.
- 4) 작은 면적을 갖는 삼각형의 기본이 되는 블록 다각형의 변을 제거한다(그림 4(b)).
- 5) 단계 2~4를 사각형이 될 때까지 반복한다.(그림 5(a)~(d))



[그림 4] FSMBR 생성 과정(초기과정)



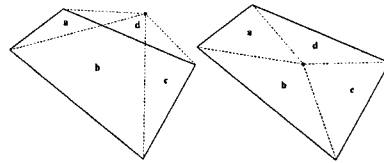
[그림 5] FSMBR 생성 과정(반복과정)

이와 같이 FSMBR 생성을 위해서는 매우 많은 단계의 작업이 필요하다. 특히 점의 개수가 많아질수록 단계의 수가 급격히 늘어나는 단점이 있다.

그래서 이 논문에서는 공간 객체 다각형의 블록화 작업 이후에 수행되므로 속도 향상 효과가 있다. 그리고 다른 근사 표현법에 비해 FSMBR이 차지하는 면적이 가장 작아 좋은 여과효율을 기대할 수 있다.

FSMBR의 PIP 알고리즘은 기존 알고리즘[4]을 사용할 수도 있지만 주어진 FSMBR이 4각형이라는 특수성을 가지므로 여과 단계의 속도를 향상시키기 위해 다음과 같은 알고리즘을 사용한다.

- 1) 사각형의 모든 변과 임의의 점 사이에 삼각형을 만든다.
- 2) 생성된 4개 삼각형(그림 6의 a,b,c,d)들의 면적의 합을 구한다.
- 3) 면적의 합이 기존 사각형보다 크면 점이 사각형 외부에 있고(그림 6(a), 그렇지 않으면 내부(그림 6(b))에 있다.

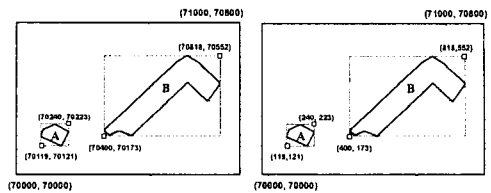


[그림 6] FSMBR의 PIP

### 3.3 최소경계 사각형의 압축 표현

#### (1) RMBR : 상대좌표값을 이용한 MBR

그림 7의 (a)는 절대 좌표값을 이용한 MBR 표현으로 16바이트가 사용되는 반면, 그림 7의 (b)와 같이 상대 좌표계를 이용할 경우, 각 X,Y 좌표를 각각 2바이트로 표현할 수 있다. 그 결과 저장공간을 절반으로 줄일 수 있다.



(a) 절대 좌표값(MBR) (b) 상대 좌표값(RMBR)

[그림 7] 절대 좌표값과 상대 좌표값 표현

#### (2) QMBR : 정량화에 의한 MBR

각 MBR의 좌표값을 정량화[5]하면 표현량을 더 줄일 수 있다(그림 8(a)). 정량화 단계가 256이하이면 각 좌표값을 1바이트로 표현가능하므로 QMBR은 4바이트의 표현량이 필요로 하여 기존 MBR에 대해 4배의 압축효과가 있다.

그러나 QMBR은 MBR의 커짐 현상이 발생하고 이는 정제 단계에 참여 하는 객체의 수가 증가하며 전체 수행 시간의 증가를 초래한다. 프로세서 성능이 매우 낮은 PDA에서는 매우 큰 단점이다.

#### (3) HMBR : MBR 크기를 고려한 혼합 표현

이 논문에서 제시하는 HMBR 표현법은 MBR의

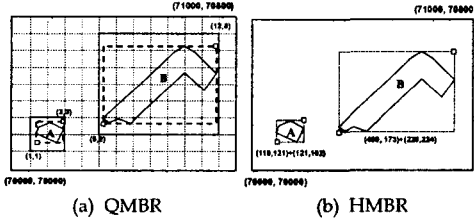
좌화점은 상대 좌표계처럼 표현하되 우상점을 MBR의 크기로 표현하는 혼합 표현법이다.

$$X_{\min}(2b) + Y_{\min}(2b) + X_{\text{크기}}(1b) + Y_{\text{크기}}(1b)$$

MBR의 크기값을 1바이트로 표현하기 위해 우선 크기값이 임계값( $\beta$ ) 이하이면 그대로 표현하고, 크기값이 임계값보다 크면 정량화된 값으로 표현하는 것이다. 이때, 임계값  $\beta$ 는 255-n이며, n은 정량화단계로서 버킷 공간을 n 등분함을 의미한다. 그리고 정량화 크기  $\delta$ 는 버킷공간/n이며, 정량화된 값은  $\beta + (\text{MBR의 크기}) / \delta$ 이다.

예를 들어 그림 8의 (b)는 절대좌표값으로 표현된 그림 7의 (a)를 HMBR로 표현한 것이다. 객체 A는 크기값을 그대로 표현한 경우이며, 객체 B는 n이 50일 때 정량화된 값으로 크기를 표현한 경우이다. 결론적으로 HMBR은 6바이트로 표현가능하고 전체적인 MBR의 표현량이 줄어들게 된다. 대부분의 공간 객체의 크기는 일정 기준치 이하이므로 여과 효율의 저하는 거의 없다.

이와 같이 MBR 압축 기법을 사용했을 때 점질의와 같은 공간 연산 수행을 위해 압축된 MBR을 복구하는 과정에서 다소의 연산이 필요하다. 그러나 여러 개의 압축된 MBR을 복구하지 않고, 오히려 질의 좌표를 변환하면 압축에 의한 부하를 간단히 줄일 수 있다.



[그림 8] QMBR과 HMBR

MBR과 이 논문에서 제안한 BMBR과 FSMBR의 장단점을 간략히 정리하면 표1과 같다. 기존 MBR은 저장 용량이 적게 차지하는 반면 여과 효율 또한 낮지만 여과 단계의 처리 속도는 매우 빠르다. 따라서 고속의 프로세서를 가진 서버 시스템이나 여과 단계를 빨리 끝내고 정제단계를 병렬로 처리하는 환경에 적합하다.

FSMBR은 저장용량 측면에서는 기존 MBR의 2배가 소요되지만 여과 효율이 제일 뛰어나 정제단계에 참여하는 객체의 수가 현저히 감소한다. 따라서 FSMBR은 복잡한 부동소수점 연산이 많은 정제단계의 복잡도에 치명적인 모바일 휴대 장치에 적용하기 좋은 근사 표현법이다. 그리고 BMBR은 적절한 저장 표현 용량과 여과 효율을 보임으로써 PC 기반 공간정보 시스템에 적절할 것으로 판단된다. HMBR은 표현 용량이 기존 MBR의 1/3이하지만, 여과 효율과 처리속도는 비슷하여 서버 시스템과 연동된 소규모의 메모리 용량을 가진 클라이언트 모바일 단말장치에 적합하다. 따라서 제시한 각 근사기법은 최근 멀티 플랫폼 기반 GIS 응용에 유

용하게 사용될 수 있다.

<표 1> 근사 표현법 장단점 분석

	여과 효율	처리 속도	표현 용량	적용시스템
MBR	저	고속	소량	서버 컴퓨터
BMBR	중	중속	중량	PC 시스템
FSMBR	고	저속	대량	stand alone mobile
HMBR	저	고속	극소	client mobile

### III. 결론

이 논문에서는 기존 MBR 이외에 기울어진 경계 사각형(BMBR)과 자유형 경계 사각형(FSMBR)을 제안하였고, 하나의 공간 객체로부터 BMBR과 FSMBR을 생성하는 방법을 자세히 다루었다. 또한 MBR이 서버 중심의 시스템에 적합하며, 여과 효율이 좋은 FSMBR은 독립형 모바일 장치의 공간 연산 처리에 적합함을 알 수 있었다.

그리고 공간 효율성을 높이기 위한 방안으로서 MBR의 크기를 고려한 혼합 표현법인 HMBR은 그 표현량을 줄이면서 공간 데이터의 대부분을 차지하는 크지 않은 객체인 경우에는 오차가 전혀 없다는 장점이 있다. 제안한 HMBR은 클라이언트 모바일 장치에 적합함을 알 수 있었다.

앞으로는 전술 바와 같이 이 논문에서 제안한 근사 표현법의 성능을 평가하기 위해 각 근사 기법 별로 여과 및 정제 단계의 수행시간, 여과 효율, 표현 용량 측면에서 실험하고자 한다. 그리고 제안한 근사 표현법을 R\*-tree에 적용하여 기존 R\*-tree와의 성능 비교를 수행하고자 한다.

### 참고 문헌

- [1] 김진덕, 최진오, "휴대폰 기반 벡터 지도 서비스를 위한 클라이언트-서버 시스템의 캐싱 기법", 한국해양정보통신학회 논문지, 6권 3호, pp. 452-458, 2002
- [2] T. Brinkhoff, H.P. Kriegel, R.Schneider, B. Seeger, "Efficient Processing of Spatial Joins Using R-trees", Proc. of Int. Conf. on Management of Data, ACM SIGMOD, pp. 237-246, 1993
- [3] 장직현, 컴퓨터 알고리즘/C++, 사이텍미디어, 2000
- [4] 이재규, C로 배우는 알고리즘, 세화, 1994
- [5] T.J. Lehman, M.J. Carey, 1986, "A Study of index structures for main memory database management system", Proc. of Int. Conf. on VLDB, pp. 294-303