공간 객체의 근사화 방법 연구

김 용 현[†] · 이 형 수[††] · 이 성 수[††] · 김 응 모[†††]

요 약

지형 정보 시스템에서 공간 객체수가 방대하여, 보조 기억 장치에 저장하는데 공간 객체의 엑세스를 빠르게 하기위해서 공간 데이터베이스 시스템을 연구하였다. 공간 객체들은 SAM으로 구성되었지만, 공간 다각형이 직접 SAM 을 구성 할수 없다. 공간 다각형을 처리하기위해 가장 대표적으로 MBR이 지형 키로서 공간 다각형 대신 사용된다. 질의 처리시 MBR은 빠르지만 부정확하다. 따라서, 공간 객체를 근사화하는데 어떤 근사화 방법이 사용되느냐가 질의 처리시 성능에 영향을 미친다. 적절한 근사화 방법이 후보 집합을 줄일수 있다. 근사화의 질이 높을수록 필요없는 엑세스를 줄일수 있다. 본 논문에서는 Slice 분리라는 다중 용기를 이용한 근사화 방법을 제안하였고 다른 근사화 방법과 비교하였다.

A Study of Approximation Method of Spatial Objects

Yong Hyun Kim[†] · Hyung Soo Lee[††] · Sung Soo Lee[†††] · Ung Mo Kim[††††]

ABSTRACT

Since the number of spatial objects go easily into millions, they are stored on a secondary storage. In order to speed up accessing the geometric objects, we investigate spatial database system. The spatial objects are organized and accessed by spatial access methods(SAMs). But, SAMs are not able to organize polygons directly. The most popular approch for handling polygon in SAMs is to use MBR apporximaotion as a geometric key. MBR provides a fast but inaccurate answers to approximation-based query processing. The performance of approximation-based spatial query processing depends on which type of approximation is chosen for the spatial objects. A suitable approximation is crucial for reducing the size of the candidate set.

The better the approximation quality, the fewer accesses to the exact object are necessary. In this paper, I proposed a new multicontainer approximation named Slice decomposition. The performance of proposed method is compared with other method.

1. 서 론

지리 공간 내에 분포하고 있는 공간 객체들을 효율적으로 입력, 저장, 관리하고 사용자의 목적에 따라 분석 처리해 주는 지형 정보 시스템-GIS(Geometric Information System)에 대한 연구가 활발히 진행되어 왔다. 공간에 존재하는 공간 객체는 공간 좌표를 포함하는 공간 데이터들과 일반적인 속성을 나타내는 비 공간 데이터들로 구성된다. 공간상에서 객체들은 점(point), 선(line), 영역(area)의 기본 객체들로 표현되며, 이런 2 차원 혹은 3 차원 공간 데이터는 1 차원 구조의 데이터를 다루는 기존의 자료 구조로는 지원이 어렵다. 또한, 수백만개가 넘는 공간 객체의 데이

† 정 회 원:현대정보기술 개발실 근무
†† 정 회 원:한국전자통신연구소 선임연구원
††† 비 회 원:성균관대학교 정보공학과 교수
논문접수:1995년 12월 29일, 심사완료:1996년 8월 27일

터 양이 매우 방대하며 각 객체마다 특성이 다르므로 같은 선이나 같은 점 객체라도 다른 크기의 데이터를 가지므로 비정형이라는 특징이 있다. 사용자가 GIS 내의 데이터를 효율적으로 얻기 위해서는 공간 질의어를 사용하는데 이 공간 질의어는 크게 기하 질의(geometric query), 위상 질의(topological query)로 나눌 수 있고, 다시 기하 질의는 위치(posi tion), 거리(distance), 가까움(nearest), 멈(furthest)의 질의로 나눌 수 있고, 위상 질의는 인접(adjacent), 겹침(cover), 교차(intersection), 포함(contain)등의 질의로 나눌 수 있다. 이런 질의는 단독으로 사용할 수 있지만, 복합적인 형태의 질의도 많이 사용된다.

GIS를 보다 효율적으로 사용하기 위해 해결해야 할 문제점으른 공간 질의 처리시 수 많은 객체가 질의에 의해 검색되고 처리되는데, 이때 방대한 자료를 보관, 처리하는 일이 지형 정보 시스템의 성능에 가장 큰 영향을 미친다고 할수 있다. 시스템의 성능을 향상시키기 위해 공간 질의시 참여하는 공간 객체의 수를 줄여 디스크 참조 횟수와 비교, 계산 횟수를 최소로 해야한다. 지형 객체를 저장하고 각 객체에 접근하기 위해서는 공간적인 성질을 포함하는 구가 필요한데 R-tree 구조가 많이 쓰이고 있다. 이 R-tree는 각 공간 객체를 지형 키(geometric key)를 이용해 인식하는데 이런 지형 키에는 대표적으로 MBR이 사용된다. 이 MBR은 간단하지만 공간 질의에 참여하는 공간 객체을 줄이는데는 효율이 좋지 못하다. 효율을 향상시키기 위해서 단순한 MBR 보다는 좀 복잡하더라도 효율이 좋은 근사화 방법이 연구되어 왔다. 본 논문에서는 여러 근사화 방법을 소개하고 기존의 방법보다 정확성을 높인 새로운 근사화 방법을 소개하겠다.

본 논문의 구성은 2장에서 지형 정보 시스템에 대해 간단히 설명하고, 3장에서는 단일 용기를 이용한 근사화 방법과 다중용기를 이용한 근사화 방법을 비교하여 분석하여보고, 4장에서는 새로 제안한 근사화 방법의 소개와 비교 실험 결과를 제시하겠다. 5장의 결론으로 끝을 맺겠다.

2. 지형 정보 시스템

2.1 데이터 층

점, 선, 다각형등 공간 객체는 최소한 하나이상의 공간 어트리뷰트(spatial attribute)와 비 공간 어트리뷰트(non spatial attribute)로 구성된다[1]. 이 공간 어트리뷰트는 점 선, 다각형과 같은형의 2차원 혹은 3차원의 데이터를 갖는다. 이때 같은 어트리뷰트로 정의된 공간 객체의 모임을 공간 관계(spatial relation)라 한다. 예를 들어 도시(CITY)와 숲(FOREST)이 다음과 같은 공간 데이터 및 비공간 데이터를 갖는다고 할 때, CITY (Cname, postal code, population, Cregion), FOREST (Fid, Fname, Fregion), Cregion과 Fregion은 공간 어트리뷰트로서 2 차원 공간 다각형이고, Cname, postal code, id, Fname등은 비 공간 어트리뷰트이며, CITY, FOREST는 각각 독립된 하나의 관계(relation)를 구성한다.

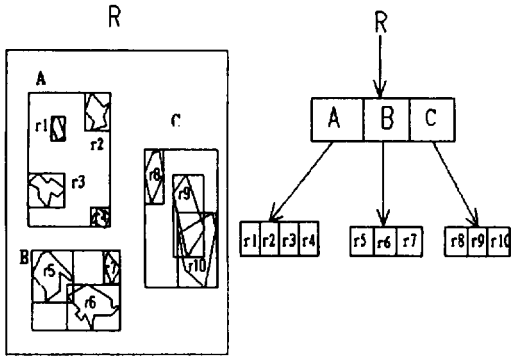
지도상에는 여러가지 지형적인 특징과 관계들이 통합되어 나타난다. 도로, 강, 호수, 건물, 도시 경계선, 전화박스등 각각의 특징들은 점, 선, 영역과 같은 기본 요소(basic primitive)들로 표현된다[2]. 도로를 나타내는 선의 굵기로 도로의 넓이를, 등고선의 색으로 높이를, 호수의 크기로 넓이를 알 수 있지만, 한 도시에서 다른 도시로 가는 가장 가까운 거리를 찾는다든지, 댐 공사시 댐의 높이에 따라 변하는 수몰지역의 넓이라든지 좀더 복잡한 분석에는 한계가 있다.

정확한 분석을 위해 GIS내의 공간 데이터(spatial data)는 하나의 지도를 주제(theme)에 따라 여러 부분 지도로 나뉘어 진다. 예를들어 도로만을 나타낸 지도라든지, 상, 하수도관이나 가스관만을 나타내는 지도 등으로 나눌수 있다. 보통 지도상에 모두 나와 있는 특징들을 먼저 기본 요소(basic primitive)로 분류하고 같은 기본 요소는 다시 세부 분류한다. 각각은 층을 이루는데 각 층마다 특징과 특징을 특성을 나타내는 어트리뷰트들로 구성된다. 각 층은 사용자의 이용 목적에 따라 구성된 관계(relation)라 할 수 있다. 각 데이터 층은 가스관을 보수하기 위한 가스관의 표시와 공사 차량의 이동을 위한 도로의 지도를 하나로하는 것과 같이 여러 층을 합쳐서 하나의 새로운 주제도(thematic map)를 구성할 수있다.

2.2. 공간 접근 방법 : SAMs

하나의 관계(relation)을 이루는 객체들에 접근하기 위해서 객체들을 지원하는 별도의 자료 구조가 필요

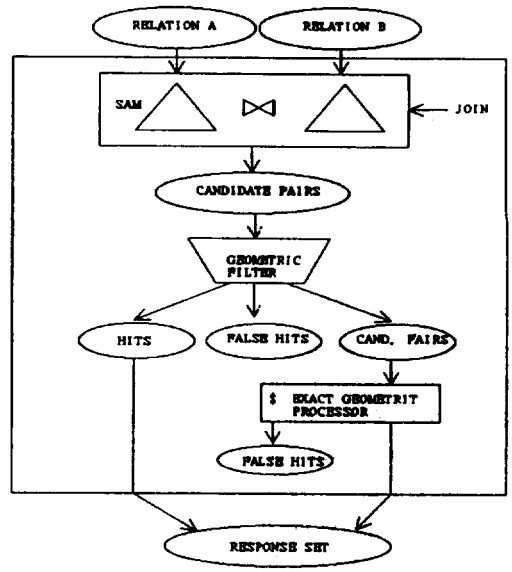
하다. 이 자료 구조는 공간 적인 성질이 도입되어 공간 연산시 공간 객체들을 효율적으로 접근하여야 한다. 공간상의 객체들은 공간상 모양과 위치를 결정하는 여러 가지 지형적인 구성 요소에 의해 특징 지워지고, 객체들은 기본 요소(basic primitive)로 표현된다. 객체를 구성하는 기본 요소에 따라 SAM은 여러 방법으로 구성된다[1, 2]. 점(point) 데이터일 때는 quad tree, k-d tree, grid file을 선, 영역 데이터일 때는 R-tree, skd tree등을 이용한다[3]. 여기서는 영역을 나타내는 다각형을 다루기 위한 인덱스 구조인 R-tree의 변형 R*-tree를 이용한다[4, 5, 6].



(그림 1) R* 트리
(Fig. 1)R*-tree

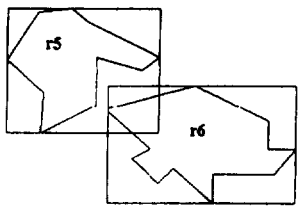
인덱스는 튜플(tuple)의 집합으로 기억장소에 기억되어 있고, 각 튜플은 공간 객체에 해당한다. R*-tree는 높이 균형 트리(height balanced tree)이고, 잎 노드와 중간 노드로 구성되며, 잎 노드와 중간 노드의 구조가 틀린데, 잎 노드의 구조는 〈rect, id〉이다. rect는 공간 객체를 포함하고 있는 직사각형(MBR)의 두 좌표 값인 LUX(Left Upper X 좌표), LUY(Left Upper Y좌표), RLX(Right Low X 좌표), RLY(Right Low Y 좌표) 값으로 나타내고, id는 r1, r2, r3 …와 같이 공간 객체의 인식자(identifier)이다. 중간 노드 루트 노드의 구조는 〈rect, ref〉이다. rect는 자식 노드(child node)내의 모든 엔트리(entry)들을 포함하는 직사각형의 두 좌표 값을 나타내고, ref는 자식 노드의 주소 값을 나타낸다. 이때, rect는 지형 키인 MBR이 사용된다. 수많은 공간 객체와 각 객체당 많은 양의 데이터를 포함하고 있는 만큼 디스크 억세스 횟수를 줄이기 위해 공간 객체들은 자신을 MBR과 같이 근사(approximation)화 시켜 인덱스의 크기를 줄이고, 근사화한 어떤 값을 키값으로 갖는다. 이때 공간 객체를 참조하기 위한 지형 키(geometric key)로 근사화한 값을 사용한다.

2.3. 공간 질의



(그림 2) 공간 질의처리기
(Fig. 2) Spatial query processor

공간 정보를 바탕으로 공간 상의 객체가 공간 질의에 만족하는지를 검사하여야 하는데 이는 Join 연산(operation)에 의해 이루어진다[7]. 객체 대신 R*-tree를 구성하고 있는 근사화한 MBR을 사용하므로 MBR에 의한 Join 연산이 이루어진다. 예를 들어, "도시와 교차하는 숲을 모두 찾아라"와 같은 질의는 위의 질의를 처리하기 위해서 Relation A를 CITY로 Relation B를 FOREST 라하면 공간 질의 처리는 아래의 그림과 같은 순서에 의해 이루어진다. CITY와 FOREST



(그림3) 잉여 후보 객체의 예

는 각각 MBR로 구성된 R*-tree에서 JOIN 연산에 의해 서로 교차하는지를 결정하게 된다. MBR에 의한 Join 연산은 공간 객체 자신이 아닌 근사화된 도형의 연산이므로 수행 결과는 우리가 얻고자 하는 결과이외의 잉여 결과를 얻게 된다[8, 9].

위의 그림은 〈r5 ,r6〉후보 쌍에 해당하는 예이다. 지형키에 해당하는 MBR은 교차하지만 각 객체들은 교차하지 않는다. Join 연산의 잉여 결과는 후보 집합에 포함되고, 다음 단계로 후보 집합을 여과(filtering)하는 단계를 거치게 된다. 대표적인 여과 기법은 false-area test 기법과 progressive approximation 기법이다[10]. false-area test 기법은 공간 객체의 면적과 근사화된 객체와의 면적 의 관계로 결정된다. 두객체의 근사화한 도형의 교차면적이 객체의 false-area 보다 크면 두 객체는 교차한다고 보는 것이다.

progressive approximation 기법은 공간 객체의 progressive approximation가 서로 교차하면 객체는 교차한다고 본다. progressive approximation의 방법에는 원이 내접하는 MEC와 직사각형이 내접하는 MER이 있는데, 두 객체 A와 B의 MECA와 MECB 또는 MERA 와 MERB가 교차한다면 공간 객체 A와 B는 서로 교차한다고 본다[8]. 위의 두 정제 방법의 조건에 일치하지 않는 객체를 제거(false hit)하고, 조건을 만족하는 객체를 질의 결과 집합에 첨가하여, 후보 집합의 객체 수를 줄이게 된다. 여과 과정에서 남은 객체들은 최종적으로 근사화된 모양이 아니라 객체 자체를 읽어들여 실질적으로 교차하는지를 검사하는 exact geometric processor 단계를 거치게 되는데 이때 저장 장소의 접근이 가장 많이 이루어지며, 전체 질의 시간중 가장 많은 시간이 소모하게 된다[9]. 따라서 후보 객체의 수를 줄이는 것이 exact geometric processor 단계를 줄이는 최선의 방법으로 공간 질의 처리의 성능을 높이는 가장 중요한 요소임을 알수 있다.

3. 근사화 방법

MBR의 Jion 연산은 실제 공간 객체가 아닌 객체를 근사화 시킨 직사각형으로 잉여 결과를 초래하게 되었다. 이 직사각형을 용기(container)라 하며, 하나의 MBR이 하나의 객체를 포함하므로 단일 용기(single container)라 한다. 앞에서의 false-area test에서 본바와 같이 근사화의 용기가 클수록 Join 연산에 의한 후보 객체의 수는 많아지게 되고 많아진 만큼 공간 질의시 참여하게 되는 객체가 많아져 질의 처리시간이 길어지게 된다. 잉여 결과는 공간 객체 자신과 근사화된 도형간의 공간상 면적의 차이에 의해 발생하게 되는데, 이 면적의 차이를 false area (approximation area-obj area)라하며, 이 false area를 최대한 줄임으로써 잉여 결과도 줄일 수 있다. 따라서, 용기의 크기를 줄여 객체와 가장 면적이 비슷한 근사화 용기를 선택하는 것이 필요하다.

다음은 잉여 결과를 줄이기 위해 연구된 근사화(approximation)기법을 소개한다. 아래의 근사화 방법은 공간 객체를 근사화된 도형 안에 모두 포함시키는 conservative approximation이라 한다.

conservative approximation은 하나의 근사화된 도형에 포함시키는 단일 용기(single container) 방식과 객체를 여러개의 용기에 나누어 포함시키는 다중 용기(multicontainer)방식이 있다 〈표 1〉[11].

〈표 1〉 보존 근사화
〈Table 1〉 conservative approxiamtion

single container	multicontainer	
	grid oriented	object oriented
MBR RMBR	ErrorBound(c)	RectlinearSizeBoun(c)
CH	SizeBound(c)	Convex decomposition
m-C	Edge Quadtree	Trapezoid decomposition
MBC MBE	(c:constant)	Triangle decomposition

3.1 단일 용기를 이용한 근사화 종류

▷ MBR(Minimum Bounding Rectangle)

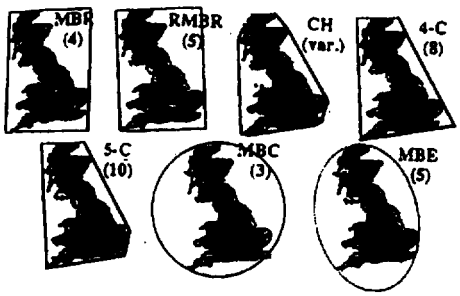
객체를 포함하는 가장 작은 직사각형으로 왼쪽 위의 점과 오른쪽 아래의 점으로 저장이 가능하다.

- parameter 수:4개
- 평균 false area:0.93

▷ RMBR(Rotated Minimum Bounding Rectangle)

MBR에 회전을 허용하여 근사화의 질을 높인 것이다. MBR의 4개와 회전 parameter을 더하여 5개 필요.

- parameter 수:5개
- 평균 false area:0.62



(그림 4) 단일 용기 근사화 방법 [12]
(Fig. 4) Single container approximation

▷ CH(Convex Hull)

객체의 모양에 따라 다른 모양의 불록하게 근사화된 모양을 이루고, parameter 갯수는 공간객체의 모양에 따라 가변적이다.

- parameter 수:4-160
- 평균 false area:0.24

▷ n-C(minimum bounding n-Corner)

근사화된 도형은 n개의 다각형 꼭지점을 갖는다.

- parameter 수:2 × n개
- 4-C:parameter 수:8개
  평균 false area:0.44
- 5-C:parameter 수:10개
  평균 false area:0.33

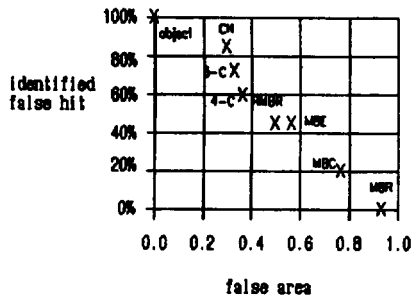
▷ MBC(Minimum Bounding Circle)

중심점인 x, y좌표를 기준으로 반경 r인 원의 모양. parameter 갯수는 3개.

▷ MBE(Minimum Bounding Ellipse)

각도에 따라 반경의 값이 다른 타원으로 중심점 의 x, y좌표와 타원을나타내는 matrix $\begin{pmatrix} A & B \\ B & C \end{pmatrix}$를 포함하여 5개 parameter를 갖는다.

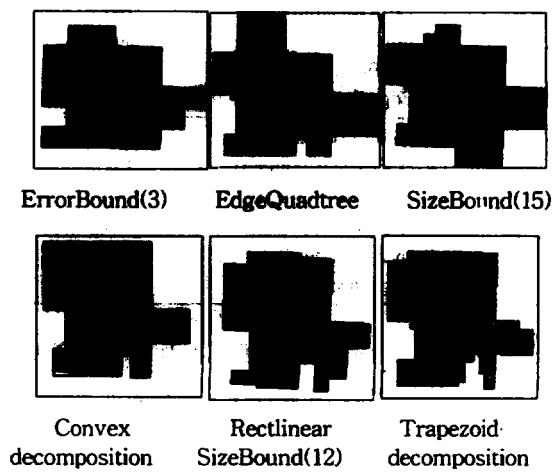
MBR Join 연산에 의해 얻은 후보 집합중 임여결과를 100으로 하고 위의 5가지 근사화 방법으로 정제한 결과 100개중에서 CH 방법은 최고 81개의 false hit를 찾아내었다. 이는 최종 exact geometric 단계에서 100개가 아닌 19개의 공간 객체만을 읽어들이므로 질의 시간이 많이 단축된다. CH 방법은 근사화를 구성하는 정점들(vertices)의 수가 매우 불규칙적이며, 다른 근사화 기법에 비해 평균 저장 공간이 매우 크므로 수많은 객체가 참여하는 공간 질의에는 부적절하다. CH에 비해 5-C는 비교적 적은 10개의 정점을 갖고, false area도 아래의 그림에서 볼 수 있듯이 CH와 차이가 거의 없다. 정제(filtering)로 사용하였을 때도 2/3 이상인 68%의 false hit를 찾아내는 성능을 보여 전체적인 성능 면으로 볼 때 가장 바람직한 근사화 방법이라고 할 수 있다. (그림 5)은 false area와 정제 성능을 비교한 것이다. 위에서 볼 수 있듯이 false area가 적은 근사화 방법일수록 성능이 좋으므로, false area와 정제 성능은 반 비례함을 알 수 있다.



(그림 5) false area와 성능 비교
(Fig. 5) Performance comparison with false area

3.2 다중 용기를 이용한 근사화

공간 객체를 하나이상의 용기들로 분리(decomposition)함으로서 single container를 사용할때보다 근사화된 도형의 면적을 최소로하며 false-area도 줄일수 있다. 아래 그림은 다중용기를 이용한 근사화 방법의 예들이다[11]



(그림 6) 다중 용기 근사화 방법
(Fig. 6) Multi-container approximation

▷ ErrorBounding(c)
- 공간 객체는 분리된 grid cell (element)로 표현 되는데 cell의 수는grid resolution 2c에 의해 결정된다.

▷ SizeBound(c)
- 근사화된 도형의 element 수를 C 개로 제한한다.

▷ Edge Quadtree
- 중심점을 중심으로 4 분원으로 나누고, 다시 재귀적으로 나뉘어 표현한다.

▷ RectilinerSizeBound(c)
- C 개이하 직사각형으로 분리(decomposition).

▷ Trapezoid decomposition
- 위에서부터 각 정점을 지나는직사각형으로 분리.

▷ Triangle decomposition
- 3개의 정점을 지나는 3각형으로 분리.

▷ Convex decomposition
- 객체를 볼록한 다각형 모양으로 분리한다.

〈표 2〉 보존 근사화의 면적 비교
〈Table 2〉 Area comparison by conservative approximation

Representation		Approximation area orea
original object		100%
single container	MBR	164%
	MBC	163%
	CH	117%
multi-container	– grid oriented –	
	ErrorBound(6)	146%
	SizeBound(5)	309%
	SizeBound(20)	156%
	Edge Quadtree	168%
	– object oriented –	
	RectlinearSizeBound(5)	158%
	RectlinearSizeBound(20)	115%
	Convex decomposition	123%
	Trapezoid decomposition	115%
	Triangle decomposition	280%

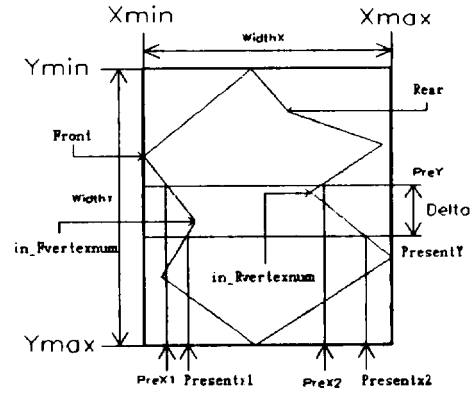
위의 표에서 볼수있듯이 공간 객체 자신의 면적을 100으로 하였을때 다른 근사화 방법들의 면적은 모두 다르게 나타난다. 앞에서 언급한바와 같이 근사화된 면적과 성능은 반 비례하므로 면적이 가장 작은 근사화 방법이 가장 성능이 좋다고 할수 있다. 적절한 근사 방법의 결정은 후보 객체를 줄이는데 중요한 요소이다.

4. 제안한 근사화 방법

4.1 연구 내용

근사화의 성능을 높이기 위해서는 근사화가 쉬워야 하고, 근사화된 도형의 정확성을 높여 false area를 최대한 줄이고 근사화 면적을 최소로 하여야 한다. 본 논문에서는 근사화 도형의 면적을 최대한 줄일수 있는 새로운 다중용기(multiple container)를 이용한 근사화 방법을 제안하였으며, 이 방법은 기존의 방법들과는 달리 면적을 최대한 줄이기 위해 객체의 x축 넓이인 WidthX와 y축 넓이인 WidthY중에서 큰 쪽의 값을 일정한 간격의 델타(delta)값으로 잘라내어 근사화 하였다.

이 방법의 장점은 첫째, 공간 객체를 분리(decomposition)하는데 있어서 다른 다중 용기를 이용한 방법들보다 간단하다고 할수있다. 그 이유는 기존의 방법은 공간 객체를 분리하는데 x축이나 y축의 크기에 상관없이 공간 객체의 점이나 면적으로만 나누어 최적화된 근사 도형을 얻는 알고리즘을 사용하여 복잡한 반면에, 새로 제안된 방법은 단순히 일정한 델타 값을 증가시키므로 단순화 할수 있다. 둘째, 델타 값을 가중치로 생각할수 있다. 다시말해서, 짧은 응답 시간을 요하고, 정확하고 세밀한 공간 객체를 원할때 공간 데이터를 구성할때 델타의 값을 최소로 명시함으로써 근사화 도형의 정확성을 높일수 있고, 처리 시간도 줄일수 있다.

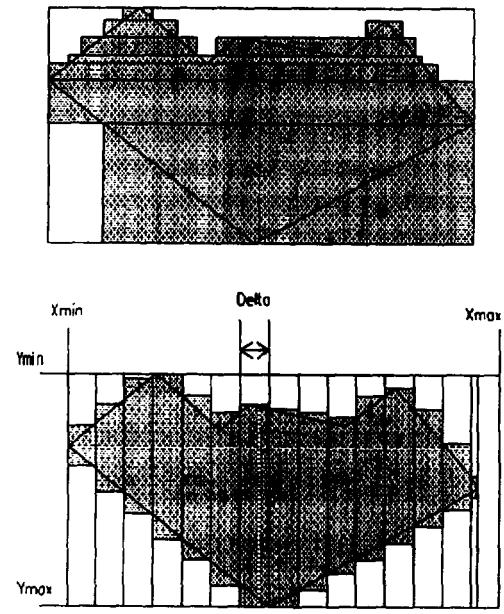


(그림 7) 각 변수의 예

PresentY는 PreY값에 델타(delta)값을 증가시켜 결정된 Y좌표 값이며, PreY는 델타(delta)값 증가 이전의 Y좌표 값이다. PreXi는 Y좌표 값을 델타(delta)만큼 증가하기 이전에 공간 객체와 교차하는 점들의 x좌표 값이며, PresentXi는 PresentY 좌표값과 교차하는 공간 객체의 점들의 x좌표 값이다. 아래는 공간 다각형의 클래스 구조이다.

```
class POLYGON
{
  private:
      int poly_num;          // 객체 번호
      int Num_Vertex;        // 꼭지점의 갯수
      int MBR[2][2];         // MBR 꼭지점 좌표
      int **Co_Vertex;       // 꼭지점의 좌표
      slice_Rectangle **slice; // slice 직사각형의
                               MBR
      int slice_Num[10];
public:
      POLYGON(void);         // 생성자
    ~POLYGON(void);          // 소멸자
      void polynum( int );   // 다각형 번호
      void func_Vertex_Longy (void);
      void func_Vertex_Longx (void);
      void func_Slice_Longy(void);
                        // Y축을 decomposition
      void func_Slice_Longx(void);
                        // X축을 decomposition
      void func_Slice_area(void);
      // slice intersection x축이 긴 경우
      friend int operator& =(POLYGON a,
      POLYGON b);
      // slice intersection y축이 긴 경우
      friend int operator| =(POLYGON a,
      POLYGON b);
};
```

임의의 공간 객체를 MBR과 다중 용기 방식중 성능이 우수한 trapezoid 방식과 위의 클래스 구조의 프랜드 함수을 이용하여 구성하여 비교해 보았다. 아래 그림의 검은 부분이 근사화 도형의 면적에 해당한다. (그림 8)은 근사화 방법중 false area의 면적이 가장 작은 trapezoid decomposition 방법과 새로 제안한 slice 방법을 비교하였다.



(그림 8) Trapezoid 분할과 slice 분할
(Fig. 8) Trapezoid decomposition and slice decomposition

4.2 실험 결과

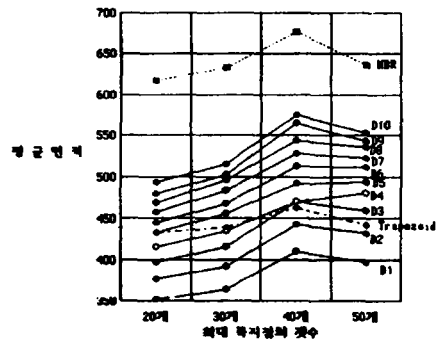
실험은 새로 제안한 slice 분리(decomposition)기법과 기존의 분리 기법인 MBR, Trapezoid를 비교 하였다. 앞절에서 언급한 바와 같이 공간 JOIN 연산시 수행시간을 단축하고, 비교를 쉽게하기 위해서 근사화 기법을 사용하는데 근사화 방법에 따라 공간 객체와 면적의 차이가 있고, 그 면적으로 인한 실질 공간 객체와 근사화 도형의 교차 횟수와 차이가 있음을 알수 있었다. 이런 교차 횟수의 차이는 저장 공간의 탐색 횟수의 증가와 비례하며, JOIN 연산의 성능에 커다란 영향을 끼친다 할 수 있다.

본 논문에서는 근사화 도형의 면적을 최소로 하는데 중점을 두었으며, 실험은 기존의 방법과 새로 제안한 slice방법을 면적에 대한 비교와 그 면적에 대한 교차 회수를 비교 하였다.

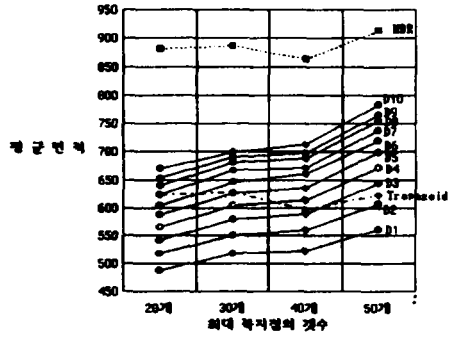
▶ 면적 비교

결과 1은 MBR의 최대 면적이 40×40 이고 최대

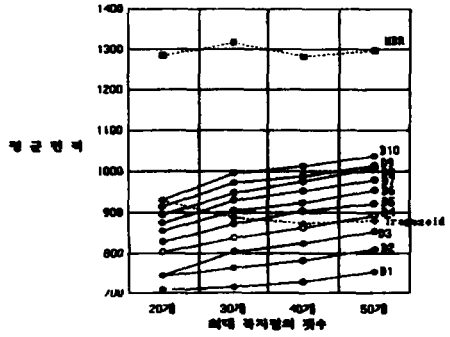
꼭지점의 갯수가 20인 다각형 100개에 대해 MBR의 평균 면적과 Trapezoid decomposition의 평균 면적과 새로 제안한 Slice방법으로 delta의 크기를 1 부터 10까지 변환하였을때 평균 면적을 비교하였고, 최대 꼭지점의 갯수가 각각 30, 40, 50인 다각형 100개에 대하여도 비교하였다. 결과 2, 결과 3은 최대 면적이 50 × 50, 60 × 60인 다각형에 대해 결과 1과 같은 방법으로 비교하였다.



결과 2 - 1



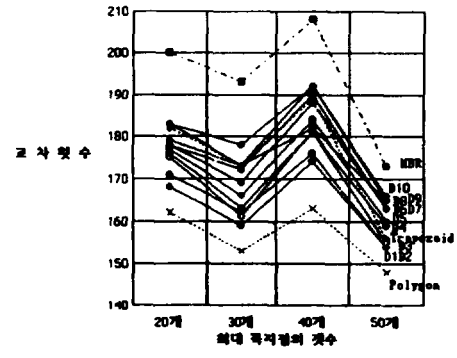
결과 3 - 1



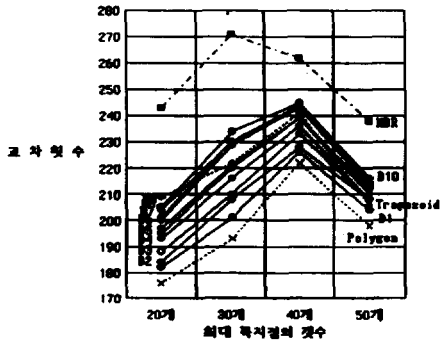
▶ 교차 횟수 비교

교차 횟수 비교의 결과는 면적 비교 결과 1-1, 2-1, 3-1과 같은 평균 면적을 갖는 다각형들의 교차 횟수를 비교한 것으로서, 각각 다각형의 갯수는 100개씩이며, MBR의 최대 면적은 40 × 40, 50 × 50, 60 × 60이고, 다각형의 최대 꼭지점의 갯수 20, 30, 40, 50개로 제한하였다. 이때, 100개 다각형 자체의 교차 횟수와 이 다각형을 근사화 시킨 다각형인 MBR, Trapezoid decomposition 방법을 사용하였을때 지형 정보 시스템 공간상에서 교차 횟수를 비교하였다. 결과 1-2의 한 예로 최대 꼭지점의 갯수가 20일때, 다각형 100개에 대하여 교차 횟수를 비교해 보면, 공간 다각형 객체 자체는 공간상에서 162번의 교차가 이루어 진데 반해, 단일 용기 방식인 MBR 근사화 방법으로 공간 객체를 구성하였을땐 200번의 교차가 이루어 졌다. 다중 용기 방법중 성능이 우수한 Trapezoid decomposition 방법으로 공간 객체를 구성 하였을때는 182회의 교차가 이루어 졌다. 새로 제안한 Slice decomposition 방법으로 공간 객체를 구성하여 delta = 1일땐 168회의 교차가 이루어졌으며, delta = 10일땐 183회의 교차 결과가 나왔다. 이는 단일 용기 방식인 MBR 보다는 Trapezoid decomposition 방식이 우수하며, Trapezoid decomposition 방식 보다는 Slice decomposition 방식 중에서 delta의 값이 작을수록 교차 횟수가 줄어드는것을 알 수 있다. 최대 면적과 최대 꼭지점의 갯수를 변화시켜 2-2와 3-2의 결과를 얻었다. 다른 결과도 1-2 같이 면적이 작을수록 교차 횟수가 줄어듬을 알수 있었다.

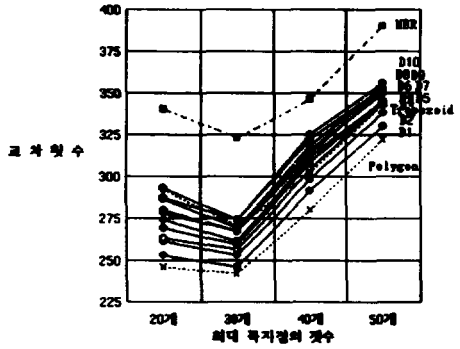
결과 1 - 2



결과 2 - 2



결과 3 - 2



5. 결 론

GIS에서 공간 객체를 근사화하여 공간 객체를 단순화 시킴으로서 처리속도를 높이고자 하였다. 그러나 근사화의 방법에 따라 false area의 차가 심하게 났고 false area가 클수록 JOIN 연산시 많은 잉여 결과로 성능은 떨어짐을 알수 있었다.

결과 1-1에서 최대 꼭지점의 갯수가 20이고, MBR decomposition의 평균 면적이 616.84일때, 다각형 100개의 MBR 교차 횟수는 200번이고, Trapezoid decomposition의 평균 면적이 432.812일때, 다각형 100개의 교차 횟수는 182번이다. Slice decomposition에서 delta = 1일때, 평균 면적은 351.56이며, 교차 횟수는 168회이고, delta = 10 일때, 평균 면적이 439.175이고, 교차 횟수는 183회이다. 즉, 결과에서 볼수 있듯이 결과의 평균 면적이 클수록 다각형의 교차 횟수도 증가함을 알수 있다.

본 논문에서는 기존의 방식과는 달리 공간 객체를 분리하는데 있어서 크기를 정하여 크기에 따라 일정하게 잘라내는 방법을 제안하였다. 따라서, 크기 즉 delta의 값을 어떻게 정하는냐가 가장 중요한 요소라 할수 있는데, delta의 값은 공간 객체의 중요성에 맞추어 정하는 것이 적당하다. delta 값을 가중치로 생각하여 세밀하고 정확한 작업을 요하는 공간 객체의 구성시에는 delta의 값을 작게하고, 일반적인 작업시에는 적절한 delta값을 정함으로서 공간 객체를 보다 효율적으로 정의할수 있다.

참 고 문 헌

[1] Stan Aronoff, "Geographic Information Systems: A Management Perspective", WDL Publications, Ottawa, Canada. 1989.

[2] Robert Laurini, Derek Thompson, "Fundamental of Spatial Information Systems", The A.P.I.C. Series Number 37.

[3] "Hierarchical Spatial Data Structure", Lecture Note in Computer Science 409, Springer-Verlag, pp. 193-2031989.

[4] Hanan Samet, "The Design and Analysis of Spatial Data Structures", Addison-Wesley PUBLISHING COMPANY, INC.

[5] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching", Proc. ACM SIGMOD International Conference on Management of Data, Boston, MA, pp. 47~57, 1984.

[6] Hans-Peter Kriegel, Ralf Schneider, Bernhard Seeger, Beckmann N., "The R*-tree: An Efficient and Roburst Access method for Points and Rectangles", Proc. ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, NJ, pp. 322-331, 1990.

[7] Thomas Brinkhoff, Hans-Peter Kriegel, Bernhard Seeger, "Efficient Processing of Spatial Joins, Using R-trees", Proceedings ACM SIGMOD International Conference on Management of Data, Washing, DC, USA, pp. 237~246, 1993.

[8] Thomas Brinkhoff, Hans-Peter Kriegel, Ralf

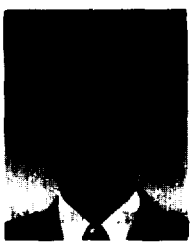
Schneider, Bernhard Seeger, "Multi-Step Processing of Spatial Joins", ACM SIGMOD Minneapolis, Minnesota, USA. pp. 197~208, 1994.

[9] Thomas Brinkhoff, Hans-Peter Kriegel, "The Impact of Global Clustering on Spatial Database Systems", Proceedings of the 20th VLDB Conference Santiago, Chile, pp. 168~179, 1994.

[10] Hans-Peter Kriegel, Thomas Brinkhoff, Ratt Schneider, "Efficient Spatial Query Processing in Geographic Database Systems", Bulletin of the Technical Committee on Data Engineering, September, VOL. 16, NO. 3, pp. 10-15, 1993.

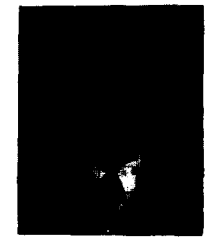
[11] Hans-Peter Kriegel, Peter Heep, Stephan Heep, Michael Schiwietz, Ralf Schneider, "An Access Method Based Query Processor for Spatial Database System", GDBMS Workshop Preceedings, Italy, pp. 273~292, 1991.

[12] Thomas Brinkhoff, Hans Peter Kriegel, Ralf Shneider, "Comparison of Approximations of Complex Objets Used for Approximation-based Query Processing in spatial Database Systems", Proc. 9th Int. Conf. on Data Engineering, Vienna, Austria, pp 40-49, 1993.



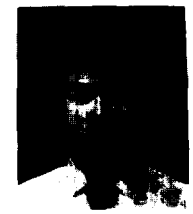
김 용 현

1993년 성균관대학교 정보공학과 졸업
1996년 성균관대학교 정보공학과 석사졸업
현재 현대정보기술 개발실 근무
관심분야: 역(deductive) 데이터베이스,지형 정보 시스템(GIS)



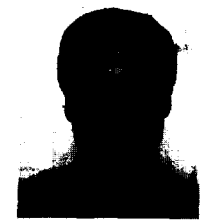
이 형 수

1980년 경북대학교 전자공학과 학사
1986년 연세대학교 전자계산학과 석사
1995년 성균관대학교 정보공학과 박사
1983년~현재 한국전자통신연구소 선임연구원
관심분야: 무선망 설계 엔지니어링, GIS 데이터처리, 컴퓨터/네트워크 보안, 데이터 통신



이 성 수

1983년 한국 항공대학교 항공전자공학과 학사
1990년 한국 항공대학교 대학원 항공 전자공학과 석사
1984년~현재 한국전자통신연구소 선임연구원
관심분야: 전파 엔지니어링, 스펙트럼관리, 개인통신, GIS



김 응 모

1981년 성균관대학교 수학과 학사
1986년 Old Dominion Univ. 석사
1990년 Northwestern Univ. 박사
1990년~현재 성균관대학교 정보공학과 교수
관심분야: 연역(deductive) 데이터 베이스, 객체지향 데이터베이스, 능동 데이터베이스, 지형 정보 시스템(GIS)