DOI: 10.3745/KIPSTD.2009.16-D.2.161

이동 객체 궤적의 색인을 위한 개선된 분할 알고리즘

전 현 준[†] · 박 주 현[††] · 박 희 숙[†††] · 조 우 현[††††]

요 약

최근 GPS, 이동 전화, 무선 네트워크 등의 발달로 인해 넓은 공간상에서 시간의 흐름에 따라 변화하는 이동 객체에 대한 위치 정보를 수집하여 실생활에 활용하는 다양한 위치 기반 서비스의 사용이 늘어나고 있다. 그와 함께 대용량의 이동 객체를 빠르게 검색하기 위한 효율적인 색인 방법의 필요성이 대두 됨에 따라 관련된 많은 연구가 현재 진행 중이다. 본 논문에서는 이동 객체의 궤적에 대한 색인 과정에서 필요한 개선된 궤적 분할 방법을 제안한다. 궤적의 적절한 분할 위치를 찾아 근사치 영역을 나타내는 최소 경계 사각형(MBR)을 만드는 과정에서 평균적인 질의의 크기를 고려하여 형성되는 확장된 최소 경계 사각형(EMBR)의 영역을 이용한다. 이에 따라 EMBR의 총면적이 최소에 가까운 분할을 만들어내어 색인 구성 후 질의 수행 과정 동안에 불필요한 탐색 공간을 감소시키는 이점을 보이게 된다. 본 논문에서 제안하는 궤적 분할 방법의 우수성을 입증하기 위해 최적의 궤적 분할 방법과 기존의 궤적 분할 방법을 구현하여 각각의 EMBR 면적을 비교 분석한다. 비교 결과 제안하는 궤적 분할 방법이 기존의 방법보다 최적의 분할에 더 가까운 EMBR의 총면적을 나타내는 것을 알 수 있었다.

키워드 : 궤적 분할, 이동 객체, 시공간 색인, 범위 질의, 이동 데이터베이스

An Improved Split Algorithm for Indexing of Moving Object Trajectories

Hyun-Jun Jeon[†] · Ju-Hyun Park[††] · Hee-Suk Park[†††] · Woo-Hyun Cho[††††]

ABSTRACT

Recently, use of various position base servicesthat collect position information for moving object and utilize in real life is increasing by the development of wireless network technology. Accordingly, new index structures are required to efficiently retrieve the consecutive positions of moving objects. This paper addresses an improved trajectory split algorithm for the purpose of efficiently supporting spatio-temporal range queries using index structures that use Minimum Bounding Rectangles(MBR) as trajectory approximations. We consider volume of Extended Minimum Bounding Rectangles (EMBR) to be determined by average size of range queries. Also, Use a priority queue to speed up our process. This algorithm gives in general sub-optimal solutions with respect to search space. Our improved trajectory split algorithm is going to derive minimizing volume of EMBRs better than previously proposed split algorithm.

Keywords : Trajectory Splitting, Moving Object,Spatiotemporal Indexing, Range Query

1. 서 론

최근GPS(Global Positioning System), 이동 전화, 무선 네트워크 등의 발달로 인하여 넓은 공간상에서 시간에 따라 변화하는 객체들에 대한 위치 정보를 수집하여 실생활에 활용하는 다양한 위치 기반 서비스의 사용이 점차 늘어나고 있다. 따라서 이들 정보들에 대한 관리를 위하여 사용하게 되는 이동 객체 데이터베이스(Moving Object Database)에 대한 관심이 증가하고 있다. 이동 객체란 시간의 변화에 따라 공간적인 위치 및 모양이 연속적으로 변하는 시공간 데이터(Spatio-temporal Data)로 이러한 이동 객체가 네트워크를 통해 서버로 자신의 위치 정보를 전송한다. 대용량의 위치 정보를 전송 받은 서버에서는 빠르고 효율적인 검색을 제공하기 위해 수집된 데이터들에 대한 적절한 색인을 구성하게 된다. 이 때에 다양한 시공간 범위 질의들을 효율적으로 수행하기 위한 목적으로 특별한 색인 구조가 필요하게 되는데 현재까지 R-tree[14]와 같은 공간 색인 구조로부터 발전된 여러 가지 색인 방법이 제안되었다[1, 2]. 대부분의 색인 방법들에서는 공통적으로 최소 경계 사각형(MBR : Minimum Bounding Rectangle)과 같은 실제 데이터에 대한 근사치 영역을 사용하여 노드를 구성한다.

효율적인 이동 객체의 색인을 위하여 제안된 여러 가지 방법들을 분류해 보면 질의의 종류에 따라 이동 객체의 이

† 준 회 원 : 부경대학교 컴퓨터공학과 공학석사
†† 준 회 원 : 부경대학교 전자컴퓨터정보통신공학부 시간강사
††† 정 회 원 : 부경대학교 전자컴퓨터정보통신공학부 시간강사
†††† 종신회원 : 부경대학교 전자컴퓨터정보통신공학부 교수
논문접수 : 2008년 5월 27일
수 정 일 : 1차 2008년 11월 17일
심사완료 : 2008년 12월 5일

력 정보를 검색하기 위한 색인 방법과 이동 객체의 궤적을 검색하기 위한 색인 방법, 그리고 이동 객체의 궤적을 분석해 미래를 예측하기 위한 색인 방법들이 존재한다. 여기서 이동 객체의 궤적이라는 것은 공간상에서 시간의 변화에 따라 움직이는 객체의 이동 경로 즉, 시간의 흐름에 따라 기록된 포인트의 순서라고 할 수 있다. 이러한 궤적의 데이터를 검색하기 위한 시공간 질의 유형은 좌표 기반 질의와 궤적 기반 질의 2가지로 구별할 수 있다[3]. 좌표 기반 질의는 주어진 시간 간격 동안 어느 공간 영역을 가로지르는 궤적의 객체 식별자와 객체의 수를 반환한다. 궤적 기반 질의는 궤적들이 서로 복잡하게 교차하는 위상적인 관계 혹은 주어진 시간 간격 동안 어느 공간 안에서의 속도 변화와 이동 방향 등의 정보를 반환하게 된다. 이러한 질의 유형들을 처리하기 위해 궤적을 분할하여 MBR들을 만들고 이를 이용하여 색인을 구성한다. 따라서 시공간 색인 과정에서 이동 객체의 궤적에 대한 분할과 MBR집합의 구성 방법이 더 나은 질의 성능을 위하여 중요한 요소가 된다. 즉, 이동 객체 궤적에 대하여 탐색 공간이 작은 색인을 한번 생성해두면 이 후에 이 궤적에 대한 수많은 질의들의 처리 시간을 줄일 수 있다. 그래서 색인의 생성 시간보다는 색인의 탐색 공간을 최소화하는 것이 매우 중요하다.

본 논문에서는 시공간적 특성을 갖는 이동 객체의 궤적을 색인 하는 과정에서 필요한 개선된 궤적 분할 방법을 제안한다. 제안하는 궤적 분할 방법은 분할 과정에서 평균적인 범위 질의의 크기를 고려하여 확장되는 최소 경계 사각형(EMBR : Extended MBR)의 면적을 이용해 분할의 위치를 찾아낸다. 최종적으로구성되는 MBR집합은 EMBR의 총면적이 최적에 가까운 최소값을 가짐으로써 질의 수행 과정 동안에 불필요한 탐색공간을 감소시키는 이점을 보이게 된다. 성능 비교 결과 제안하는 궤적 분할 방법이 기존의 선형 시간 궤적 분할 방법(Linear Time Trajectory splitting)[4]보다 최적의 분할이 가지는 EMBR의 총면적에 더 가까운 값을 가진다는 것을 알 수 있었다. 그리고 분할의 위치를 찾기 위한 계산 과정에서 우선순위 큐(Priority Queue)[11]를 사용하여 계산에 소요되는 시간을 단축하였다. 또한 본 논문에서 제안하는 방법은 저장 공간에 제한이 있을 경우 그에 적절한 분할을 만들기 위하여 분할 개수의 한계를 두는 방법으로 확장이 용이하다.

본 논문의 나머지 구성은 다음과 같다. 2절에서는 이동 객체의 시공간 색인과 궤적의 분할 방법에 대한 관련 연구들을 살펴보고 3절에서는 EMBR의 총면적이 최소가 되는 최적의 궤적 분할 방법을 설명한다. 4절에서는 기존의 궤적 분할 방법보다 더 최적에 가까운 분할을 위한 개선된 궤적 분할 방법을 제안한다. 5절에서는 제안하는 궤적 분할 방법의 우수성을 입증하기 위해 기존 방법과의 성능 평가를 수행한다. 마지막으로 6절에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

이동 객체의 위치를 검색하기 위해 주로 쓰이는 공간 색인 방법으로는 격자 파일(Grid File)[12], 사분 트리(Quad Tree)[13]와 같은 공간 분할 색인과 R-tree[14]와 같은 데이터 분할 색인이 있다. 그 중에서 현재까지 제안되고 있는 시공간 색인 구조들을 살펴보면 대부분 R-tree[14]를 기준으로 발전된 방법들을 많이 사용하고 있다. 이동 객체의 색인을 위하여 제안된 색인 방법들은 질의의 종류에 따라 이동 객체의 이력 정보 탐색을 위한 색인 방법[5, 7-9], 이동 객체의 궤적 탐색을 위한 색인 방법[3, 6, 16, 17, 21], 그리고 이동 객체의 미래 위치를 예측하여 탐색하기 위한 색인 방법[19, 20]으로 나눌 수 있다. 그 중 이동 객체의 궤적 탐색을 위한 색인방법들은 공간적인 차원과 시간적인 차원을 처리하는 방법에 따라 세가지로 분류할 수 있다.

첫 번째 분류는 시간이 추가된 하나의 공간적인 차원으로 간주된 경우이다[3, 6]. TB-tree[3]에서는 기존 R-tree[14]에 시간을 하나의 공간적인 차원으로 확장하여 만든 색인 방법으로 이동 객체 궤적을 보존하기 위한 양방향 연결 리스트를 이용함으로써 전체 또는 부분을 탐색하는 질의를 효율적으로 처리할 수 있다. 그러나 R-tree의 노드 간에 겹침 현상이 많이 일어난다는 단점이 있다. 3DR-tree[6]는 객체가 존재하는 궤적에 대한 MBR의 면적이 커짐으로써 빈 공간을 많이 포함하기 때문에 효율적인 색인이라 할 수 없다. 두 번째 분류는 시간과 공간이 결합된 스키마 안에서 서로 다르게 처리되는 경우이다[16, 17]. 하지만 SETI-tree[16]와 SEB-tree[17]는 색인 과정에서 MBR을 사용하지 않는다. 세 번째 분류는 시간을 공간으로부터 다르게 처리하는 경우이다[21]. OP-tree[21]에서는 이동객체의 궤적에 더욱 가까운 근사치 영역을 만들기 위해 MBR의 각 모서리를 잘라 팔각형의 영역(Octagon Prism)을 만드는 것을 제안하였다. 그러나 MBR을 사용하는 R*-tree와의 성능 비교 결과 뛰어난 이점이 있다는 것을 입증은 하지 못하였다.

이러한 방법들은 이동 객체의 궤적에 대한 색인 구성 과정에서 다양한 유형의 질의에 대하여 적절한 응답을 수행하기 위해 MBR과 같은 실제 데이터에 대한 근사치 영역을 사용한다. 그러므로 색인 구성과정에서 근사치 영역 MBR을 어떻게 만들어 내는지가 검색 효율을 위해 중요한 요소가 된다. 효율적인 색인의 구성을 위해 이동 객체 궤적을 분할하여 MBR을 만들어 내는 기존의 방법으로는 합병 분할 방법[10]이 있고, 가장 최근에 제안된 방법으로서는 EMBR의 면적을 고려하여 선형적으로 분할을 만들어 내는 방법[4]이 있다.

합병 분할 방법[10]에서는 이동객체의 궤적에 대한 근사치 영역이 포함하는 빈 공간을 최소화 하기 위하여 근접한 MBR들을 묶어서 분할 위치를 찾는 알고리즘을 제안하였다. 그리고 분할 수의 최대 한계치를 두어서 제한된 저장 공간

에 적절한 분할을 만들 수 있다는 장점이 있다. 그러나 범위 질의가 주어졌을 때에 질의의 크기에 따라 비교 검색해야 할 EMBR의 영역 문제는 분할과정에서 고려하지 않았다. 그로 인해 여러 가지 다양한 유형의 범위질의에 대해서 효율적인 분할을 만들어 내지 못한다는 단점이 있다.

선형 시간 궤적 분할 방법[4]에서는 측정 시간 구간의 왼쪽에서 오른쪽으로 선형적으로 진행해 가며 궤적을 분할하는 알고리즘을 제안하였다. 이 방법은 객체가 계속적으로 갱신되는 상황에 있어서 시간적으로 큰 장점을 갖는다. 그러나 저장 공간의 제한이 있을 경우 분할 개수의 한계를 두기 위한 방법으로는 적당치 못하다. 그리고 실험결과 질의의 범위가 커짐에 따라서 EMBR의 총면적이 커지는 현상이 발생하여 비효율적인 분할이 된다는 단점이 있다는 것을 알 수 있었다.

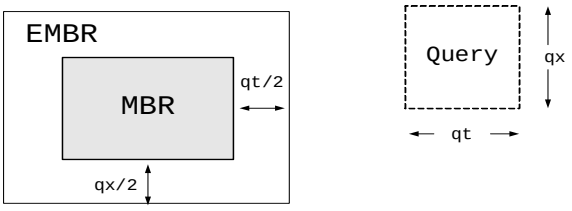
3. 최적의 궤적 분할

최적의 궤적 분할이란 이동 객체의 궤적에 대하여 범위 질의의 크기를 고려한 각EMBR의 총면적이 가장 최소가 되는 분할을 말한다. 실험 및 성능평가를 위해 선형 시간 궤적 분할 방법[4]과 본 논문에서 제안하는 궤적 분할 방법과의 성능 비교 평가를 위한 기준으로 최적의 궤적 분할을 구현한다.

원래 궤적의 위치 정보는 2차원 좌표로 표현된다. 또한 시간을 고려하여 3차원으로 궤적을 나타낸다. 그러나 본 논문에서는 간단히 하기 위해 위치 정보를 1차원 좌표로 표현하고 시간 정보를 고려하여 궤적을 2차원 공간으로 나타낸다. 2차원 공간의 궤적을 3차원 공간으로 확장하는 것은 단순한 문제이다.

EMBR면적의 계산은 (그림 1)과 같이 범위질의의 가로 $qt/2$, 세로 $qx/2$ 만큼 MBR에서 확장된 사각형의 면적을 계산하면 된다. EMBR은 MBR의 질의 수행 과정에서 범위 질의가 주어졌을 때 질의가 근사치 영역 MBR과 중첩하는지 여부를 찾아내기 위하여 비교 검색해야 하는 부분을 나타내는 것이다. 그러므로 질의의 중심점에서 각 면까지의 길이만큼 MBR의 각 면에서부터 확장시킨 영역을 나타낸다.

본 논문에서는 궤적에 대한 시간 측정 구간 i 에서 j 까지 영역을 MBR로 구성한 $M[i,j]$ 의 EMBR면적을 $E[i,j]$ 로 표기한다. 그리고 포인트 P_0 에서 P_n 까지 궤적에 대한 분할 개수 l 개의 MBR집합을 $M_l[0,n]$ 로 EMBR의 총면적을 $E_l[0,n]$ 로 표기한다. l = 1 일 때는 첨자 1을 생략하여 표기한다. 즉, $M_1[i,j]$ = $M[i,j]$ 이고 $E_1[i,j]$ = $E[i,j]$ 이다.



(그림 1) 범위질의의 크기를 고려하여 확장된 EMBR

Algorithm-1 : Optimal Split for a Trajectory

Input: A set of Trajectory's Points(P_0, P_1, ..., P_n), Query size.

Output: A set of MBRs that cover the trajectory.

1) **For each** l ($1 \le l \le n$)
Compute sum of EMBRs volume such that following holds:
$$E_l[0,n] = \min_{0 \le j < n}(E_{l-1}[0,j] + E[j,n])$$
2) Select minimal sum of EMBRs volume
3) Construct MBRs corresponding to each j

(그림 2) EMBR 총면적이 최소가 되는 최적의 분할 알고리즘

시간 t 와 위치 x 의 두 축으로 표현된 2차원 공간에 시간의 흐름에 따라 측정된 이동 객체 P 의 위치 포인트들로 이루어진 궤적 있다. 여기서 가로 qt 와 세로 qx 크기의 범위 질의 q 를 고려하여 EMBR의 총면적이 최소가 되는 최적의 분할을 만든다. 그 첫 번째 과정으로 $1 \le l \le n$ 을 만족하는 분할 개수 l 개의 MBR집합 $M_l[0,n]$ 들에 대하여 $E_l[0,n]$ 을 계산한다. 이 때의 각각의 분할 위치는 j 로 둔다. 두 번째 과정에선 각각의 $E_l[0,n]$ 들을 비교하여 가장 최소값을 가지는 하나를 찾아낸다. 마지막 과정으로 찾아낸 $E_l[0,n]$ 의 분할 위치 j 를 이용하여 MBR집합 $M_l[0,n]$ 를 구성한다. 이 과정을 알고리즘으로 표현하면 (그림 2)와 같다.

[정리 1] Algorithm-1의 시간 복잡도는 $O(n \cdot 2^n)$ 이다.

증명.

위치 포인트P0 에서 Pn 까지 분할 가능한 모든 경우의 수는 2^{n-1} 개 이다. 각 분할에 대한 EMBRs 면적의 평균 연산 횟수는 $n/2$ 이다. 2^{n-1} 개의 EMBR 총면적들 중에서 최소값을 찾기 위한 연산의 횟수는 2^{n-1} 이다. 그래서 총 연산 횟수는 $(n/2) \cdot 2^{n-1} + 2^{n-1}$ 이다. 그러므로 이 알고리즘의 시간 복잡도는 $O(n \cdot 2^n)$ 이다.

4. 개선된 궤적 분할

본 절에서는 이동 객체의 궤적에 대한 색인 과정에서 적절한 분할 위치를 찾아 MBR들을 만들기 위한 개선된 궤적 분할 방법을 제안한다

4.1 궤적분할 과정

공간을 나타내는 평면상에 P_0~P_n까지의 이동 객체 위치 포인트들로 이루어진 궤적이 기록되어 있다. 먼저 기본적인 최소 영역을 MBR들로 분할한 분할 개수 n 개의 (그림 3)과

같은 완전 분할(Full Split) MBR집합 $M_n[0,n]$을 만든다.

이 때에 각 MBR들의 분할 시작위치를 i라고 한다면 $M_n[0,n]$에서의 각 MBR은 $M[i,i+1]$로 구성된다. 이와 같은 완전 분할 MBR집합을 색인 과정에서 사용 할 수도 있지만 그럴 경우 질의 수행 과정 동안 범위 질의에 대한 EMBR의 면적이 서로 겹치는 부분을 다수 포함하게 된다. 그 결과 불필요한 탐색 공간의 증가로 인하여 비효율적인 분할이 된다. 본 논문에서 제안하는 방법은 다음과 같은 과정을 수행하여 EMBR이 서로 겹치는 부분을 최소화한다.

첫 번째 과정으로 (그림 3)과 같은 완전 분할 MBR집합에서 $0 \le i < n$을 만족하는 궤적의 시간 측정 구간 i와 $i+1$로 구성된 각각의 $M[i,i+1]$들에 대하여 (그림 4)와 같이 오른쪽에 인접한 MBR들과 합병하여 만들어지는 $M[i,i+2]$의 EMBR면적 $E[i,i+2]$을 계산한다. 그리고 합병한 MBR의 EMBR 면적 $E[i,i+2]$에 합병하기전의 EMBR 면적인 $E[i,i+1]$와 $E[i+1,i+2]$의 합을 빼서 그 결과값을 키 값($KEY_{(i,i+2)}$)으로 하여 우선순위 큐[11]에 삽입한다. 삽입 된 키 값에는 연산에 사용 되었던 합병된 MBR인 $M[i,i+2]$에 대한 시간 측정 구간 i와 $i+2$의 정보가 들어있다.

여기서 우선순위 큐[11]란 일반적인 선입선출(First In First Out) 방식의 큐와는 달리 큐의 엔트리에 우선순위를 부여하여 높은 우선순위를 가진 엔트리가 먼저 큐에서 빠져나오도록 상위에 랭크 시키는 정렬 데이터 구조이다. 본 논문에서 제안하는 방법에서는 최소가 되는 값이 최고의 우선순위를 가지게 된다. 따라서 우선순위 큐에 삽입되는 키 값이 더 작은 음의 수 일수록 합병된 MBR이 더 작은 EMBR 면적을 가진다는 것으로 우선순위의 상위에 랭크 되고, 반대로 더 큰 양의 수를 가질수록 면적이 더 커지는 것으로 하위에 랭크 된다.

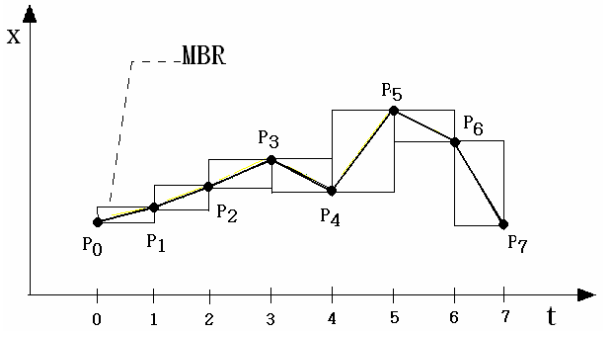
두 번째 과정에선 우선순위 큐에 제일 상위에 랭크 된 키 값을 반환하여 키 값에 들어있는 두 개의 위치 정보 i와 j를 가져온다. 그리고 i와 j로 만들어지는 $M[i,j]$를 사용하여 분할 개수가 하나 줄어든 새로운 MBR집합 $M_{n-1}[0,n]$로 갱신한다. 마지막으로 반복문의 다음 루프에서 갱신에 사용될 합병된 MBR을 찾기 위해 우선순위 큐를 재구성한다. 재구성 과정에서는 먼저 갱신에 사용된 $M[i,j]$의 왼쪽과 오른쪽에 인접한 두 개의 MBR을 각각 합병하여 만들어지는 MBR의 EMBR 면적을 계산하여 키 값 $KEY_{(i-1,j)}$와 $KEY_{(i,j+1)}$을 계산한다. 그리고 계산된 두 개의 키 값을 우선순위 큐에 추가하고 기존에 있던 같은 시간 측정 구간에 중복되는 키 값은 삭제한다. 여기까지의 두 번째 과정은 음의 수를 가지는 키 값이 더 이상 나오지 않을 때까지 반복 수행한다. 한번의 수행 과정이 진행될 때마다 EMBR의 총면적이 최대로 감소하는 한 개의 합병된 MBR이 선택되어 MBR집합의 갱신에 사용된다.

(그림 5)는 (그림 3)과 동일한 궤적의 위치 포인트를 생성하여 개선된 궤적 분할 방법을 사용해 분할되는 과정을 시뮬레이션 한 결과이다. 사각형 실선은 MBR이며 점선은 EMBR의 영역을 나타낸다.

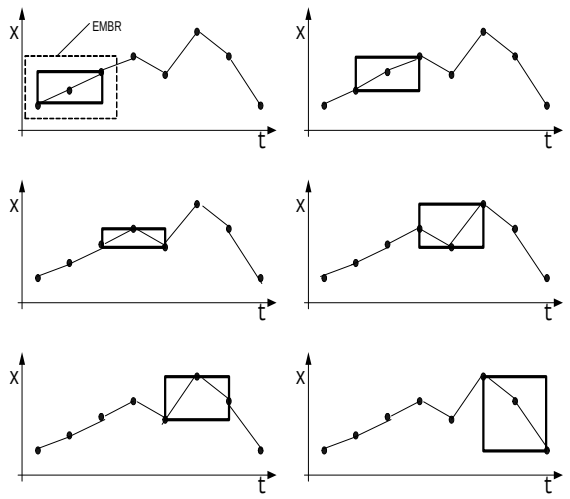
4.2 궤적분할 알고리즘

앞 절에서 설명한 개선된 궤적 분할 방법을 알고리즘으로 나타내면 (그림 6)과 같다.

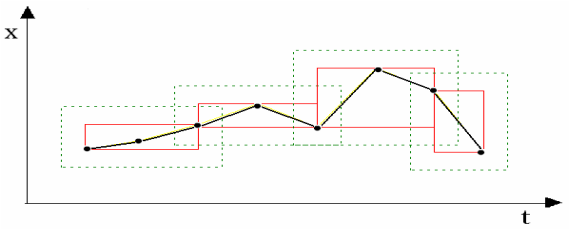
(그림 6)의 알고리즘을 간략히 설명을 하면 우선 P_0에서 P_n 까지 위치 포인트들로 이루어진 궤적과 평균적인 질의의 크기가 주어진다. 첫 번째 과정 1)에서 전체 분할 MBR집합을 만들고 두 번째 과정 2)에서 키 값을 계산해서 차례대로 우선순위 큐에 삽입한다. 다음 세 번째 과정 3)에서는 정렬된 우선순위 큐에서 첫 번째로 반환되는 키 값 $KEY_{(i,j)}$을 이용해 MBR집합을 갱신하고 우선순위 큐를 재구성한다. 음의 수를 가지는 키 값이 더 이상 존재하지 않을 때까지 3)의



(그림 3) 완전분할(Full Split) MBR 집합



(그림 4) 인접한 두 개의 MBR을 병합하여 만든 새로운 MBR



(그림 5) 개선된 궤적 분할 방법으로 구성된 최종 MBR집합

Algorithm-2 : Improved Split for a Trajectory

Input: A set of Trajectory's Points(P_0, P_1, ..., P_n), Query size.

Output: A set of MBRs that cover the trajectory.

1)Construct full split MBRs $M_n[0,n]$

2) For each i $(0 \le i < n-1)$

Compute EMBRs volume for merging $M[i,i+1]$ with $M[i+1,i+2]$

Store $KEY_{(i,i+2)}$ in a priority queue such that following holds:

$KEY_{(i,i+2)} = E[i,i+2] - (E[i,i+1] + E[i+1,i+2])$

3)While ($KEY_{(i,j)}$ in the root node of the priority queue is negative)

Use the priority queue to merge the pair of consecutive $M_l[0,n]$ that give the biggest decrease in EMBR volume;

Update the priority queue with the new merged MBR;

(그림 6) 개선된 궤적분할 알고리즘

과정을 반복 수행한 후 최종적인 MBR집합을 만들어 낸다.

[정리 2] Algorithm-2의 시간 복잡도는 $O(n\log n)$ 이다.

증명.

1)에서 전체 분할 MBR집합을 만들기 위한 연산 횟수는 n이고, 2)에서 하나의 키 값을 계산하여 우선순위 큐에 삽입하는 연산 횟수는 $1+\log n$이며 총 n번 수행이 되므로 $n+n\log n$이다. 3)에서 MBR집합의 갱신과 우선순위 큐의 재구성을 위한 연산 횟수는 $n\log n$이다. 그래서 알고리즘의 총 연산 횟수는 $n+n+n\log n+n\log n$이다. 그러므로 이 알고리즘의 시간 복잡도는 $O(n\log n)$이다.

분할의 개수를 k개로 제한하는 궤적 분할 알고리즘은 (그림 6)의 알고리즘을 이용하여 (그림 7)과 같이 간단히 설계할 수 있다. 색인 과정에서 저장 공간의 제한이 있을 경우 이 방법은 유용하다.

(그림 6)의 알고리즘을 통해 구성된 l개의 분할이 k개보다 더 작다면 (그림 7)의 알고리즘은 수행될 필요가 없다. 하지만 (그림 6)을 통해 만들어진 분할의 개수 l이 k보다 많다면 우선순위 큐에 정렬된 첫 번째 요소의 키 값을 이용하여 MBR집합의 갱신과 우선순위 큐의 재구성 작업을 l이 k가 될 때까지 계속 수행한다. 이 과정에선 MBR의 합병으로 인한 EMBR 면적의 변화가 최소로 증가되는 MBR묶음을 갱신작업에 사용한다.

Algorithm-3 : Split with limited k splits for a Trajectory

Input: A set of Trajectory's Points(P_0, P_1, ..., P_n), Query size.

Output: A set of MBRs with k splits that cover the trajectory.

1) Compute a set of MBRs using **Algorithm-2**

2) **While** ($l > k$)

Use the priority queue to merge the pair of consecutive $M_l[0,n]$ that give the smallest increase in EMBR volume;

Update the priority queue with the new merged MBR

$l = l - 1$

(그림 7) 분할 개수를 k개로 제한하는 궤적분할 알고리즘

[정리 3] Algorithm-3의 시간 복잡도는 $O(n\log n)$ 이다.

증명.

첫 번째 단계 1)에서 Algorithm-2의 연산 횟수는 $n+2n\log n$이다. 두 번째 단계 2)에서 분할 개수 l이 k가 될 때까지의 MBR집합의 갱신과 우선순위 큐의 재구성을 위한 연산 횟수는 $n\log n$이다. 그래서 총 연산 횟수는 $n+3n\log n$이다. 그러므로 이 알고리즘의 시간 복잡도는 $O(n\log n)$이다.

5. 실험 및 성능 비교

본 논문에서 제안하는 개선된 궤적 분할 방법의 성능실험은 펜티엄-IV 3.2GHz 프로세서와 메모리1.5Gbyte, Windows 운영체제를 사용하는 시스템상에서 수행되었으며, 알고리즘 구현을 위해 C#언어를 사용하였다. 그리고 성능 비교 평가를 위하여 기존의 선형 시간 궤적 분할 방법[4]과 3절에서 설명한 EMBR의 총면적이 최소가 되는 최적의 궤적 분할을 함께 구현하여 실험을 하였다. 본 실험에서는 측정 시간과 위치 정보로 표현된 2차원의 좌표상에서 위치 축의 총 범위는 0에서 1500으로 가정하였고 시간 축의 총 범위는 0에서 1050으로 가정하였다. 그리고 실험 데이터의 객관성을 위하여 한계 조건 내에서 각 데이터를 임의 생성하여 실험에 적용하였다. 실험에서 각 수치에 대한 단위는 실제 데이터 적용 시 결정되므로 여기서는 고려하지 않는다.

첫 번째 실험에서는 규칙적인 시간의 간격에 따라 변화하는 이동 객체의 위치 정보 30개를 생성하였다. 생성 과정에서 위치 변화의 정도는 각각 15, 30, 50으로 한계를 설정하였다. 그리고 10-60까지 범위 질의의 크기를 변화시켜가며 각 분할 방법을 통해 만들어지는 MBR집합의 EMBR 총면

적(Sum of EMBRs)을 계산하여 비교하였다.

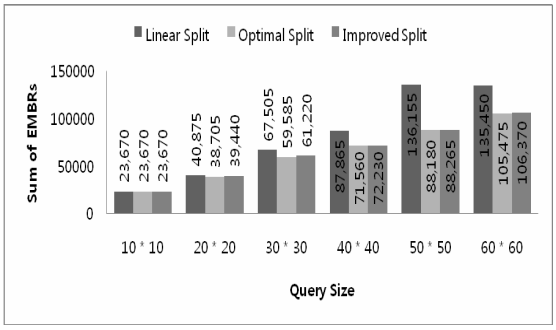
(그림 8-10)는 이동 객체의 위치 변화 정도를 다양하게 하며 실험한 각 궤적 분할 방법들의 성능 비교 결과이다. 성능 비교 결과를 보면 본 논문에서 제안하는 개선된 궤적 분할 방법을 통해 만들어지는 MBR집합의 EMBR 총면적이 최적의 분할이 가지는 EMBR 총면적과 거의 근사한 값들이 나온다는 걸 알 수 있다. 반면에 선형 시간 궤적 분할 방법[4]을 통해 만들어지는 MBR집합은 질의의 크기가 커질수록 더 큰 EMBR 총면적을 가진다는 것을 알 수 있다.

두 번째 실험에서는 시간 구간을 10-35 사이에서 변화를 주어가며 불규칙적으로 이동 객체의 위치 정보를 생성하여 실험에 적용하였다. 그 외의 조건은 첫 번째 실험과 같다.

(그림 11-13)은 두 번째 실험을 통한 각 궤적 분할 방법의 성능 비교 결과이다. 역시 첫 번째 실험과 마찬가지로 제안하는 궤적 분할 방법을 통해 만들어지는 MBR집합의 EMBR 총면적이 선형 시간 궤적 분할 방법[4]보다 더 최적의 분할에 가까운 값을 가진다는 것을 알 수 있다.

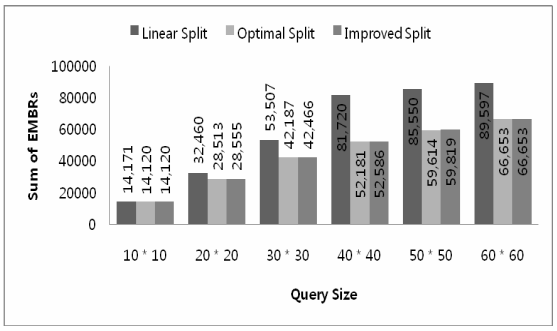
세 번째 실험에서는 객체의 포인트의 개수를 100개에서 500개까지 늘려가면서 선형 시간 궤적 분할 방법[4]과 제안하는 개선된 궤적 분할 방법과의 분할에 소요되는 시간을 측정하는 실험을 수행하였다. 성능 비교 결과는 (그림 14)와 같다.

(그림 14)의 실험결과에서 나타나는 것과 같이 위치 포인트의 개수가 많아 질수록 계산시간이 점점 차이를 보이긴 하였으나 거의 근접한 계산시간이 걸리는 것을 알 수 있다. (그림 2)의 최적의 궤적 분할 (Optimal Split) 알고리즘은 위치 포인트가 100개 일 때 0.8초의 계산 시간을 나타내었고 포인트의 개수가 많아 질수록 지수함수의 증가율을 보이므



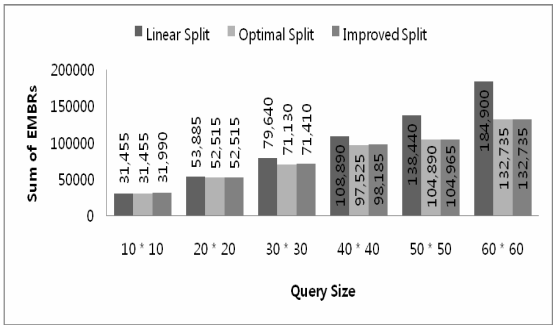
(위치변화 정도 15 이내, 측정시간 구간 35)

(그림 8) 위치정보에 대한 측정시간이 규칙적인 경우



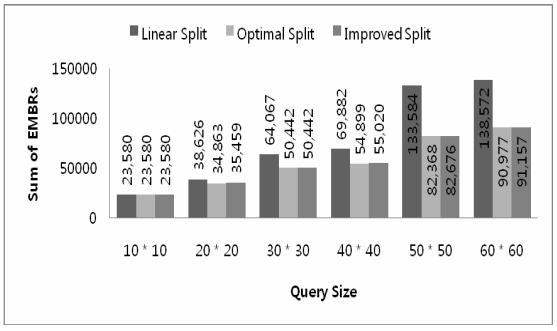
(위치변화 정도 15 이내, 측정시간 구간 10-35)

(그림 11) 위치정보에 대한 측정시간이 불규칙적인 경우



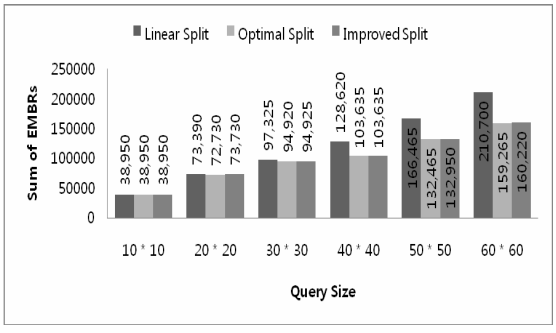
(위치변화 정도 30 이내, 측정시간 구간 35)

(그림 9) 위치정보에 대한 측정시간이 규칙적인 경우



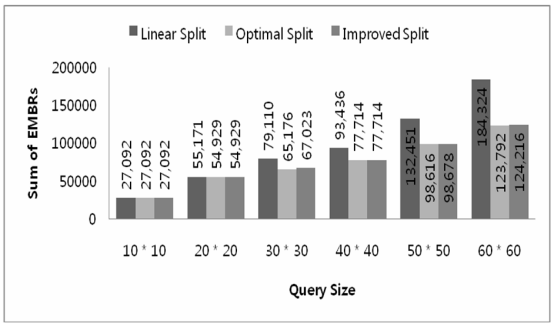
(위치변화 정도 30 이내, 측정시간 구간 10-35)

(그림 12) 위치정보에대한 측정시간이 불규칙적인 경우



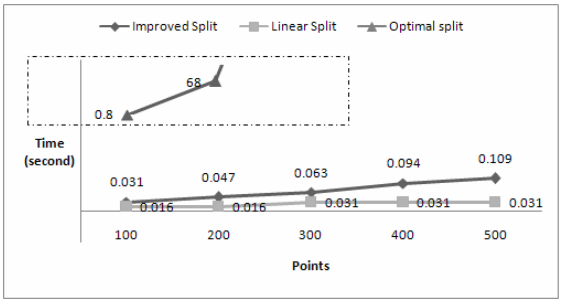
(위치변화 정도 50 이내, 측정시간 구간 35)

(그림10) 위치정보에 대한 측정시간이 규칙적인 경우



(위치변화 정도 50 이내, 측정시간 구간 10-35)

(그림 13) 위치정보에 대한 측정시간이 불규칙적인 경우



(그림 14) 포인트의 개수에 따른 궤적분할 계산시간 비교

로 그 이상의 계산 시간 측정은 생략한다.

6. 결론 및 향후 연구

본 논문에서는 이동 객체의 궤적에 대한 색인 과정에서 사용하게 되는 근사치 영역 MBR집합을 만들기 위해 필요한 개선된 궤적 분할 방법을 제안하였다. 제안하는 궤적 분할 방법은 평균적인 질의의 크기를 이용한 EMBR의 면적을 우선 고려하여 궤적의 적절한 분할 위치를 찾는다. 또한 계산 시간을 단축시키기 위해 EMBR면적의 비교 과정에서 우선순위 큐를 사용한다. 성능 비교 결과 제안하는 궤적 분할 방법은 기존의 선형 시간 궤적 분할 방법[4] 보다 더 최소에 가까운 EMBR의 총면적을 가지는 것을 알 수 있었다. 그리고 EMBR의 총면적이 가장 최소가 되는 최적의 분할에 거의 근접하는 결과를 나타내었다. 결과를 종합해보면 제안하는 궤적 분할 방법을 이용하여 이동 객체의 궤적에 대한 색인을 구성하여 둔다면 질의 수행 과정에서 불필요한 탐색 공간을 최소화함으로써 처리하는 데이터의 양이 감소되어 성능 향상을 이룰 수가 있다는 것을 알 수 있다.

향후 연구는 색인 구성 과정에서 분할된 근사치 영역을 저장할 메모리의 용량이 제한되어 있을 경우 분할 개수에 한계를 두는 알고리즘에 대한 실험을 수행할 예정이다. 그리고 현실 세계에 더욱 가까운 비용모델을 제시하기 위하여 다수의 이동객체 궤적들을 분할하는 방법에 관한 연구가 필요하다.더 나아가 제안하는 개선된 궤적 분할 방법을 적용한 공간 색인 구조의 성능 실험 또한 필요하다.

참 고 문 헌

[1] M. F. Mokbel, T. M. Ghanem and W. G. Aref, “Spatio-Temporal Access Methods,” Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, pp.40-49, 2003.

[2] D. L. Lee, J. Xu, B. Zheng and W. C. Lee, “Data Management in Location-Dependent Information Services,” IEEE Pervasive Computing, pp.65-72, 2002.

[3] D. Pfoser, C. S. Jensen and Y. Tehodoridis, “Novel Approaches to the Indexing of Moving Object Trajectories,” Proc of The 26thInternational Conference on Very Large Data Bases. pp.395-406, 2000.

[4] S. Rasetic, J. Sander, J. Elding and M. A. Nascimento, “A Trajectory Splitting Model for Efficient Spatio-Temporal Indexing,” Proc of The 31st International Conference on Very Large Data Bases. pp.934-945, 2005

[5] X. Xu, J. Han and W. Lu, “RT-Tree: An Improved R-Tree Indexing Structure for Temporal Spatial Databases,” Proc of the International Symposium on Spatial Data Handling, pp.1040-1049, 1990.

[6] Y. Theodoridis, M. Vazirgiannis and T. Sellis, “Spatio-Temporal Indexing for Large Multimedia Applications,” Proc of the 3rd IEEE International Conference on Multimedia Computing and Systems, pp.441-448, 1996.

[7] O. Wolfson, S. Chamberlain, S. Dao and L. Jiang, “Location Management in Moving Objects Databases,” Proc of the International Workshop on Satellite-Based Information Services, pp.7-14, 1997.

[8] T. Brinkhoff, “Generating Network-Based Moving Objects,” Proc of the International Conference on Scientific and Statistical Database Management, pp.253-255, 2000.

[9] R. Benetis, C. S. Jensen, G. Karciauskas and S. Saltenis, “Nearest Neighbor and Reverse Nearest Neighbor Queries for Moving Objects,” Proc of the International Symposium on Database Engineering and Applications, pp.44-53, 2002.

[10] M. Hadjieleftheriou, G. Kollios, D. Gunopulos and V. J. Tsotras, “Efficient indexing of Spatiotemporal Objects,” Proc of The 8th International Conference on Extending Database Technology, pp.251-268, 2002.

[11] D. E. KNUTH, “The Art of Computer Programming, Volume 3, Sorting and Searching Second Edition,” Addison Wesley Longman, 1998.

[12] J. Nievergelt, H. Hinterberger and K. C. Sevcik, “The Grid File: An Adaptable, Symmetric Multikey File Structure,” ACM Transactions on Database Systems, pp.38-71, 1984.

[13] T. Tzouramanis, M. Vassilakopoulos and Y. Manolopoulos, “Overlapping linear quadtrees: a spatio-temporal access method,” Proc of the 6th International Symposium on Advances in Geographic Information Systems, pp.1-7, 1998.

[14] A. Guttman, “R-trees: A Dynamic Index Structure for Spatial Searching,” Proc of the ACM International Conference on Management of Data, pp.47-57, 1984.

[15] Y. Theodoridis, R. Silva and M. Nascimento, “On the Generation of Spatiotemporal Datasets,” Proc of the 6th International Symposium on Spatial Databases, pp.147-164. 1999.

[16] Z. Song and N. Roussopoulos, “SEB-tree: An Approach to Index Continuously Moving Objects,” Proc of the 4th International Conference on Mobile Data Management, pp.340-344, 2003.

[17] V. P. Chakka, A. C. Everspaugh and J. M. Patel, “Indexing Large Trajectory Data Sets With SETI,” Proc ofthe 1st

Biennial Conference on Innovative Data Systems Research, 2003.

[18] S. Saltenis, C. S. Jensen, S. T. Leutenegger and M. A. Lopez, "Indexing the Positions of Continuously Moving Objects," Proc of the ACM International Conference on Management of Data, pp.331-342, 2000.

[19] S. Prabhakar, Y. Xia, D. V. Kalashnikov, W. G. Aref and S. E. Hambrusch, "Query Indexing and Velocity Constrained Indexing: Scalable Techniques for Continuous Queries on Moving Objects," IEEE Transactions on Computers, pp.1124-1140, 2002.

[20] Y. Tao, D. Papadias and J. Sun, "The TPR*-Tree: An Optimized Spatio-Temporal Access Method for Predictive Queries," Proc of the 29th International Conference on Very Large Data Bases, pp.790-801, 2003.

[21] H. Zhu, J. Su, O. Ibarra, "Trajectory Queries and Octagons in Moving Object Databases," Proc of the ACM International Conference on Information and Knowledge Management, pp.413-421, 2002.

전 현 준

e-mail : tgcv@pknu.ac.kr
2006년 학점은행제 컴퓨터공학(공학사)
2008년 부경대학교 컴퓨터공학과(공학석사)
관심분야 : 시공간 데이터베이스 인덱싱

박 주 현

e-mail : holy4001@empal.com
2000년 경남대학교 컴퓨터공학과(공학사)
2002년 경남대학교 컴퓨터공학과(공학석사)
2007년 부경대학교 컴퓨터공학과(공학박사)
2005년~현 재 부경대학교 전자컴퓨터정보통신공학부 시간강사
관심분야 : 공간 데이터베이스, 공간 데이터에 관한 인덱싱 기법, 객체지향 데이터베이스

박 희 숙

e-mail : bg007@edunet4u.net
1998년 경남대학교 교육대학원(교육학석사)
2006년 부경대학교 일반대학원(공학박사)
2004년~현 재 부경대학교 전자컴퓨터정보통신공학부 시간강사
관심분야 : 데이터베이스 인덱싱 성능개선 문제, RFID응용 기술, 이동객체 궤적분할 인덱싱기법에 관한 연구

조 우 현

e-mail : whcho@pknu.ac.kr
1985년 경북대학교 전산공학(공학사)
1988년 경북대학교 전산공학(공학석사)
1998년 경북대학교 전산공학(공학박사)
1988년~1989년 한국전자통신연구원 지능망연구실 연구원
2002년~2003년 텍사스주립대-산마르코스 방문연구원
1989년~현 재 부경대학교 전자컴퓨터정보통신공학부 교수
관심분야 : 지능형 데이터베이스, 멀티미디어 인덱싱, 객체데이터베이스 관리 기술