시공간 데이타베이스에서 다차원 시퀀스 데이타의 선택도추정

(Selectivity Estimation for Multidimensional Sequence Data in Spatio-Temporal Databases)

신 병 철 † 이 종 연 ††
(Byoung-Cheol Shin) (Jong-Yun Lee)

요 약 선택도 추정 기법은 질의 최적화를 위해 현재 상용 데이타베이스에서 많이 사용되고 있고 히스토그램은 가장 많이 사용되는 선택도 추정 기법중의 하나이다. 최근에 시공간 데이타베이스 관련 연구들에서 이러한 선택도 추정 기법이 기존의 시간·공간 데이타베이스 선택도 추정 기법을 확장하여 활발하게 연구되었다. 하지만 기존의 시공간 데이타베이스 선택도 추정 연구는 주로 이동 객체와 같은 시계열 데이타만 고려하였다. 또한 기존의 연구는 과거시점부터 현재 시점까지 시간적 범위 질의에 대한 선택도 추정은 불가능 하였다. 따라서 본 논문에서는 시공간 데이타베이스에서 과거 시점에서 현재시점까지 시퀀스 데이타의 시간적 범위 질의를 위한 히스토그램을 구축하고 이를 이용한 효과적인 선택도 추정 기법을 제안한다. 제안한 히스토그램을 이용하면 과거부터 현재까지 시퀀스 데이타의 선택도 추정이 가능하고, 범위 시간 선택도 추정 기법이 가능하며 효과적인 히스토그램 유지 기법의 적용이 가능하다.

키워드 : 선택도 추정, 시공간 데이타베이스, 시퀀스 데이타, 히스토그램

Abstract Selectivity estimation techniques in query optimization have been used in commercial databases and histograms are popularly used for the selectivity estimation. Recently, the techniques for spatio-temporal databases have been restricted to existing temporal and spatial databases. In addition, the selectivity estimation techniques focused on time-series data such as moving objects. It is also impossible to estimate selectivity for range queries with a time interval. Therefore, we construct two histograms, CMH (current multidimensional histogram) and PMH (past multidimensional histogram), to estimate the selectivity of multidimensional sequence data in spatio-temporal databases and propose effective selectivity estimation methods using the histograms. Furthermore, we solve a problem about the range query using our proposed histograms. We evaluated the effectiveness of histograms for range queries with a time interval through various experimental results.

Key words : Selectivity estimation, Spatio-temporal Databases, Sequence Data, Histogram

1. 서 론

건물이나 도로와 같은 특정 지형은 생성되는 시점에 초기 공간정보를 가지며 삭제되는 시점에서 현재 시점의 스냅샷 데이타베이스로부터 정보가 삭제된다. 이러한 특징을 가지는 데이타를 시퀀스 데이타(Sequence Data)라고 하며 시공간 데이타베이스에서 이동 객체와 같은 시계열 데이타와 함께 실생활에서 많이 사용된다. 최근에 시공간 데이타베이스에 대한 연구가 활발해지면서 효율적인 시공간 데이타의 저장기법과 질의 처리 비용을 감소시키는 최적화 연구들이 발표되었다. 시공간 질의는 과거, 현재에 저장되어 있는 이력정보를 검색하는 것과 미래의 객체정보를 예측하는 것으로 나눌 수 있다. 보통 과거와 현재 질의는 시퀀스 데이타와 시계열 데이타 모두에 쓰이고 미래 예측은 시계열 데이타에서만 쓰인다. 예를 들어 질의 "Retrieve all objects within query_extent at a timestamp t."는 객체 정보가 이미 데이타베이스에 존재하고 질의된 시간과 공간 영역을 만족하는 객체들을 찾는 것이다. 시퀀스 데이타들은 보통

· 이 논문은 2006년도 충북대학교 학술지원사업의 연구비 지원에 의하여 연구되었음
† 정 회 원 : 에어포인트(주) 기술연구소 연구원
suemirr@airpoint.co.kr
†† 종신회원 : 충북대학교 컴퓨터교육과 교수
jongyun@chungbuk.ac.kr
(Corresponding author임)
논문접수 : 2006년 7월 25일
심사완료 : 2006년 11월 10일

정보 변화량이 많지 않기 때문에 주된 관심이 과거와 현재의 객체 정보의 검색에 맞춰져 있다. 반면에 시계열 데이타들은 시간에 따른 정보 변화가 매우 많기 때문에 미래 시간 동안의 객체 변화 예측에 관심이 있다.

두 시계열 데이타 분야의 공통된 특징은 시간 지원에 따른 대량의 객체 정보를 갖는 것이다. 객체 정보량의 증가는 질의응답 시간이 증가하는 결과를 가져오기 때문에 효과적인 질의 처리를 위한 최적화 기법이 매우 중요하게 된다. 선택도 추정은 일반적인 데이타베이스에서 검증된 효과적인 질의 최적화 기법으로써 히스토그램 기반 기법[1-3]과 웨이블릿 기반 기법[4-7]이 있다. 또한 웨이블릿 기법 자체의 성능 향상을 위한 연구[8-10]가 발표되었다. 최근에는 기존의 선택도 추정에 관한 연구들을 기반으로 시계열 데이타들을 위한 선택도 추정에 관한 연구[11-15]가 국내외에서 활발히 진행되었다.

시공간 데이타베이스에서 히스토그램 기반 선택도 추정 기법은 내부적으로 버킷을 사용하기 때문에 이러한 버킷을 효율적으로 저장해야 한다. 이에 관련된 대표적인 연구로서 [16]이 발표되었다.

1.1 연구동기

최근에 발표된 시공간 데이타베이스에서의 선택도 추정에 관한 연구[11-15]는 모두 시계열 데이타에 한정되어 있고 시퀀스 데이타에 대한 선택도 추정 연구는 거의 전무하다. 대부분의 연구들이 이동객체와 같은 시계열 데이타를 위한 선택도 추정 연구에만 초점이 맞춰져 있기 때문에 시공간 데이타베이스에서 시퀀스 데이타를 위한 선택도 추정 기법이 필요하다. 또 다른 문제는 과거와 현재 시점에서의 시공간 선택도 추정을 처음으로 제안한 [17]에서는 특정 시점에서의 근사 질의응답만이 가능하고 범위 시간을 가지는 선택도 추정은 가능하지 않았다. 따라서 전 세계적으로 시퀀스 데이타의 범위 시간 시공간 선택도 추정에 관한 연구는 존재하지 않는다.

1.2 연구 내용 및 기여도

본 논문에서는 시공간 데이타베이스에서 시퀀스 데이타를 위한 선택도 추정 기법으로 히스토그램을 제안한다. 제안된 히스토그램은 CMH(Current Multidimensional Histogram)와 PMH(Past Multidimensional Histogram)로 나누어지고 각각 현재 시점과 과거 시점에서의 시간적 영역 질의(Range query with a time interval)를 위한 선택도 추정을 담당한다. 또한 시간의 변화에 따른 효과적인 히스토그램의 유지 기법을 제안하고 논리적인 실험을 통해 제안한 히스토그램이 시공간 데이타베이스에서 시퀀스 데이타를 위한 효과적인 선택도 추정 기법임을 증명한다.

제안된 히스토그램의 학문적인 기여는 크게 세 가지로 요약 된다: (i) 과거부터 현재까지 시퀀스 데이타의 선택도 추정이 가능하고, (ii) 범위 시간 선택도 추정 기법이 가능하고, (iii) 효과적인 히스토그램 유지 기법의 적용이 가능하다. 본 연구는 기존에 존재하지 않은 시공간 데이타베이스 선택도 추정의 연구 분야이므로 이 분야에서의 새로운 패러다임이 제시된다. 또한 기존 연구에서 해결하지 못했던 과거부터 현재시점까지 내의 범위 시간 선택도 추정 문제를 해결함으로써 선택도 추정 연구 범위를 확장시킬 수 있다.

이 논문의 구성은 다음과 같다. 제 2장에서는 본 연구와 관련된 공간 데이타베이스에서의 선택도 추정 기법인 Minskew와 시공간 데이타베이스에서의 선택도 추정 연구들에 대해 기술한다. 제 3장에서는 시공간 데이타베이스에서 시퀀스 데이타의 선택도 추정을 위한 전체적인 시스템에 대해 기술하고, 제 4장과 제 5장에서 현재 데이타와 과거 데이타를 각각 다루는 히스토그램인 CMH와 PMH에 대해 기술한다. 제 6장에서는 CMH와 PMH를 이용한 선택도 추정 기법에 대해서 기술하고, 제 7장에서는 제안한 히스토그램들의 성능 평가를 위해서 다양한 실험 평가 항목을 통해 성능을 평가한다. 마지막으로 제 8장에서는 본 논문의 결론 및 향후 연구과제에 대해 기술한다.

2. 관련 연구

2.1 Minskew

Minskew[1]는 공간 선택도 추정을 위한 히스토그램이다. Minskew 히스토그램은 버킷 분할을 통해 편중된 객체들의 분포를 균일하게 만든다. 버킷 분할 기준은 객체들의 편중도 skew에 기반하고 있으며 분할 가능한 경우의 수 중에 분할되는 두 버킷의 편중도의 가중치가 가장 낮은 분할을 선택하여 이진 공간 분할(BSP)을 한다. 각 축별로 분할 가능한 모든 경우를 검사하기에는 많은 복잡도를 가질 수 있으므로 분할 적합 축을 우선 선택한 후에 그 축을 기준으로 분할을 하는 것으로 분할 알고리즘의 효과를 높이고 있다. $B_i.num$은 i번째 버킷에 포함되는 점 객체의 수 또는 사각형 객체의 중심점이 버킷의 영역에 포함되는 수를 저장한다. C는 셀을 가리키며 $C_i.den$은 i셀에 겹치는 객체수를 저장한다. Avg(den)은 셀의 평균 밀도수 den의 평균을 말하고 |C|는 셀의 수를 가리킨다. 이 때 버킷의 편중도 $B_i.skew$는 식 (1)과 같이 표현하며 전체 편중도의 가중치는 식 (2)로 표현한다. 최종 분할 경우의 선택은 가중치가 가장 작은 것으로 한다.

$$B_i.skew=\frac{1}{|C|}\sum_{i=1}^{number\ of\ cell\ in\ Bi}(C_i.den-Avg(den))^2 \quad (1)$$

$$Minskew = \sum_{i=1}^{n} (B_i.num \times B_i.skew) \quad (2)$$

그림 1은 Minskew 히스토그램에서 버킷 분할의 예를 보이고 있다. 그림 1(a)를 보면 Dimension 1의 분산은 6이고 Dimension 2의 분산은 2임을 알 수 있다. 이것은 분산이 높은 축이 객체의 편중도가 높다는 것이고, 따라서 이러한 편중도를 낮추기 위하여 편중도가 높은 축을 기준으로 공간을 분할하는 것이 좋다는 것을 미리 알 수 있다. 그림 1(b)는 선택된 분할 축을 기준으로 가능한 분할 경우의 수에서 식 (1)을 이용하여 분할된 두 버킷의 편중도 skew를 구하고 식 (2)를 이용하여 최종 가중치를 구한 뒤 가중치가 가장 작은 분할 선을 기준으로 공간을 두개의 버킷으로 분할한 것이다. 식 (2)를 이용하여 각 분할 가능한 경우의 가중치를 구하면 첫 번째 경우의 가중치는 28.14가 나오며 두 번째 경우는 37.38이 나온다. 따라서 첫 번째 분할이 편중도를 낮출 수 있는 방법이고 그 결과는 그림 1(b)와 같다.

2.2 이동 객체를 위한 시공간 선택도 추정

[11]에서는 이동하는 객체가 어떤 시간 동안 고정된 질의 영역에 겹칠 수 있는지 여부에 초점을 맞추고 있다. 선택도 추정을 위하여 우선 전체 공간을 Minskew 알고리즘을 사용하여 버킷으로 분할하고 각 버킷에 대한 선택도 추정을 한 후 이를 모두 합하여 전체 공간에 대한 선택도를 추정하는 기술을 제시하고 있다. 2차원 공간 선택도 추정은 각 차원 별로 질의와 이동 객체를 사상시켜 1차원 환경에서 선택도를 추정한 뒤 각 차원 별로 구해진 선택도를 곱하여 2차원 공간에서의 선택도 추정을 한다. 따라서 [11]에서 제시하는 선택도 추정 기술은 2차원 공간에서는 겹치지 않는 객체가 1차원 공간으로 사상하여 겹칠 수 있는 가능성을 가지기 때문에 다차원 공간에서의 선택도 추정에 좋지 못한 성능을 보인다. 만약 객체의 속도가 [0, V]에 존재하고, 공간적으로 [0, U]내에 균일하게 분포되어 있다면 질의는 T 시간동안 질의의 공간 영역과 겹쳐지는 객체들을 찾아내는 것이다. 이 때 실제적인 선택도 Sel은 [12]에서 수식 (3)과 같이 제시하고 있다. 이 때 q_R은 사각 질의를 나타내고 2차원 공간에서 각 축별로 하한과 상한 값을 가진다.

$$Sel = \frac{VT}{U^2}[(q_{R.xmax} - q_{R.xmin}) + (q_{R.ymax} - q_{R.ymin})] + \frac{(q_{R.xmax} - q_{R.xmin})(q_{R.ymax} - q_{R.ymin})}{U^2} \quad (3)$$

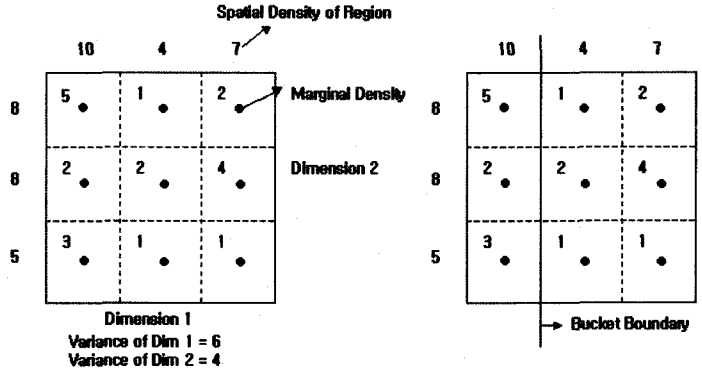
이 때 [11]에서 제시한 방법으로 선택도 추정을 한다면 각 차원 별로 식 (4)와 같은 선택도 추정이 가능하고 이를 곱하면 식 (5)와 같아지므로 선택도가 원래보다 높게 추정이 됨을 알 수 있다. 식 (4)에서 q_{Ri}는 질의의 한 차원을 가리키며 상한과 하한을 가지고 있다.

$$Seli = \frac{q_{Ri+} - q_{Ri-}}{U} + \frac{VT}{2U} \quad (4)$$

$$Sel = \frac{VT}{U^2}[(q_{R.xmax} - q_{R.xmin}) + (q_{R.ymax} - q_{R.ymin})] + \frac{(q_{R.xmax} - q_{R.xmin})(q_{R.ymax} - q_{R.ymin})}{U^2} + \frac{V^2 T^2}{4U^2} \quad (5)$$

그림 2는 [12]에서 밝힌 수식을 통한 오류율을 그림으로써 간략하게 표현한 것이다. 객체 P는 현재 시간에서의 위치 P(0)에서 다음 위치 P(1)로 이동하는 동안 질의 영역과 겹치지 않음에도 불구하고 각 차원 별로 사상하였을 경우 x축과 y축 모두 약간의 겹침 영역이 발생함을 알 수 있다.

[12]에서는 [11]에서 제시한 이동 객체 표현을 2차원 공간으로 확장시켰다. 그리고 [11]에서의 초과 선택도 추정에 대하여 2차원 공간을 1차원 공간으로 사상시키



(a) 원본 히스토그램 (b) 버킷 분할 기준

그림 1 Minskew 히스토그램의 공간 분할

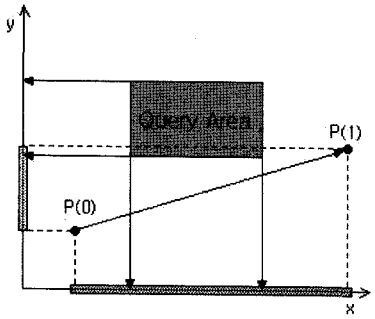
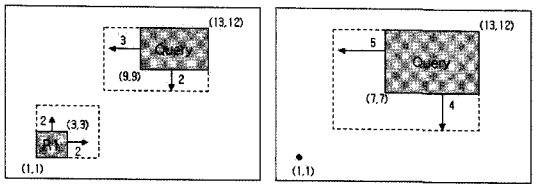


그림 2 2차원 공간에서 초과된 선택도 추정

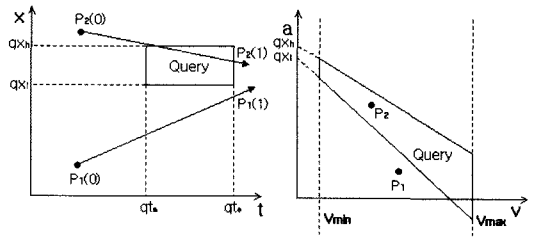
는 것을 회피함에 따라 선택도가 높아지는 현상을 해결하였다. 어떤 객체의 현재 시간 0에서의 위치를 (x, y)라 하고 각 축에 따른 속도를 (v_x, v_y)라 할 때 [12]는 Minskew 기술을 사용하여 4차원 점 (x, y, v_x, v_y)을 표현할 수 있는 4차원 히스토그램을 생성하였다. 이 때 히스토그램의 각 버킷은 영역 MBR과 속도 MBR인 VMBR을 가지며 버킷안의 객체들은 영역 MBR과 VMBR안에서 균일하게 분포되도록 히스토그램을 구축한다. 이동하는 사각형 객체와 질의에 대해서는 그림 3과 같이 객체들을 간소화하는 기술을 사용하여 풀이된다. 히스토그램의 재구축은 전체 데이타집합에서의 갱신률이 정해진 임계치를 초과할 경우에만 일어나므로 재구축 빈도가 [11]에 비하여 줄어들었다.



(a) (b)

그림 3 이동 객체의 간소화 과정: (a) 이동 객체와 질의; (b) 이동 객체의 간소화

[13]에서는 공간-시간 그래프를 Hough 변환을 통한 속도-절편 그래프로 변형한 뒤 MinSkew 기술을 적용하여 이동 객체에 대한 선택도를 추정한다. 공간-시간 그래프에서 점의 이동에 따른 궤적은 속도-절편 그래프에서 하나의 점으로 표현될 수 있으며 질의 사각형은 어떤 공간 영역으로 표현된다. 만약 속도-절편 그래프에서 점이 질의 영역에 포함된다면 공간-시간 그래프에서 그 점에 해당하는 점의 궤적인 선은 질의 사각형을 지나간다고 할 수 있다. 그림 4는 공간-시간 그래프와 속도-절편 그래프의 예를 보인다. [13]에서의 선택도 추정은 점 객체들이 존재하는 전체 공간을 절편-속도 그래프에서 MinSkew 알고리즘을 사용하여 버킷들로 분해한 뒤 질의 영역과 겹치는 버킷들의 영역을 계산함으로써 구할 수 있다.



(a) (b)

그림 4 Hough 변환을 사용한 그래프 변환: (a) 공간-시간 그래프; (b) 속도-절편 그래프

[14]에서는 클러스터링 기술을 기반으로 한 버킷 분할 히스토그램을 제시하고 있다. 이전까지의 이동객체를 위한 선택도 추정의 연구에서 대부분 Minskew 히스토그램을 확장한 데 반하여 더 효율적인 공간 버킷 분할을 위해 클러스터링 기술을 이용했다. 클러스터링 기술에서는 비슷한 성질을 가지는 객체는 가까운 거리에 존재한다는 것을 가정한다. 다시 말해 두 객체의 거리가 가깝다면 초기 위치와 속도, 객체 크기가 비슷할 수 있다. 이동 객체간의 거리 계산을 위해 유클리드 계산법을 확장하여 적용하였으며 기본적인 클러스터링 방법을 위해 그림 5와 같은 Gonzalez Clustering 방법을 사용한다. 히스토그램의 정제를 위해 이미 구축된 히스토그램의 버킷에서 어떤 객체를 뽑아 다른 버킷에 넣어봄으로써 더 나은 히스토그램이 되게 하는 기법도 제안하고 있다.

시계열 데이타를 위한 선택도 추정 기법 중 최초로 과거 시점에 대한 선택도 추정 방법을 제시한 [15]에서

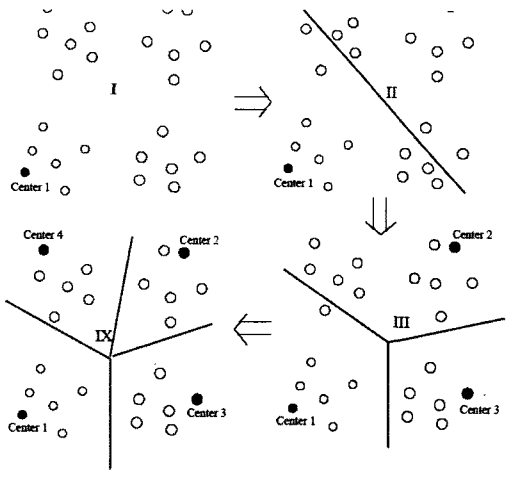


그림 5 Gonzalez Clustering

는 이동 객체의 성질에 의해 대량으로 생성되는 스트림 데이타를 효과적으로 처리할 수 있는 히스토그램 AMH (Adaptive Multidimensional Histogram)을 제안하였다. 과거와 현재 시점에서의 선택도 추정은 AMH를 직접적으로 이용하고, 미래 시간에 대한 선택도 추정은 현재와 최근 과거 정보들을 기반으로 미래 시간에서의 질의 결과를 예측하는 *exponential smoothing*에 기반 한 확률적 기법 제안을 구현하였다. 히스토그램 버킷이 갱신될 때 과거의 버킷은 주기억장치 색인에 유지되고 색인의 크기가 주기억장치의 크기를 넘어설 경우 가장 이전의 색인 부분을 디스크에 옮겨 디스크 I/O에 비용을 감소시켰다. 그림 6은 AMH에 대한 전체적인 개요이다.

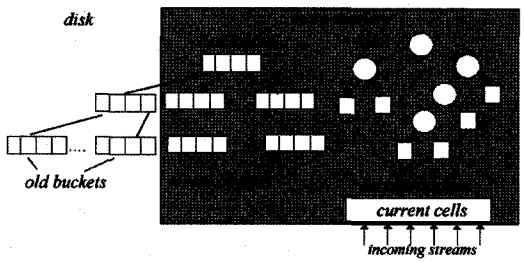


그림 6 AMH(Adaptive Multidimensional Histogram)

2.3 시퀀스 데이타를 위한 선택도 추정

시공간 데이타베이스에서 시퀀스 데이타를 위한 최초의 선택도 추정 연구 [17]에서는 공간 데이타베이스 선택도 추정기법인 Minskew 히스토그램([1])을 기반으로 시공간 히스토그램 T-Minskew로 확장하였다. T-Minskew는 히스토그램이 객체 분포의 균일함 가정을 만족하지 못하게 될 때 히스토그램 재구축을 통해 선택도 추정 오류율을 일정 수준으로 유지한다. 히스토그램 내의 각 버킷들은 히스토그램 유지 시간동안 해당 버킷 내에서 발생하는 객체 변화에 대한 정보를 시간 속성과 함께 모두 저장한다. 재구축된 히스토그램의 이전에 존재하던 히스토그램은 생성으로부터 재구축에 의한 종료 시간까지의 유효 시간이 할당되고 이력 정보로써 저장된다. 히스토그램 재구축 시기는 히스토그램 유지 동안 전체 객체 수에 비한 객체 변화 횟수가 주어진 임계치를 넘어설 경우에 발생한다.

T-Minskew의 선택도 추정 과정은 주어진 질의에 대해 먼저 질의 시간 속성 값을 포함하는 히스토그램을 찾고 질의 공간 속성 값과 겹치는 모든 버킷들을 검색한다. 선택도 추정을 위하여 검색된 버킷들의 집합에서 각각 질의 시간과 겹치는 객체 정보를 이용한다. 이 때 선택도 추정을 위한 질의는 특정 시점에서 공간 영역을 가지는 질의로 제한되어 있어 시간적 범위 질의가 불가능하였다.

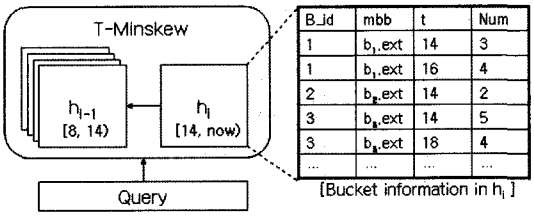


그림 7 T-Minskew의 개요

그림 7은 T-Minskew의 전체적인 개괄을 보이고 있다.

3. 선택도 추정 시스템 개요

우리는 실세계의 시공간 영역에 우리가 제안하는 기술을 적용하기 위하여 작업 공간(workspace)을 2D 공간과 시간으로 가정하였다. 현재 살아있는 시공간 객체들과 과거에 유효했던 객체들은 시간적 성질이 다르기 때문에 본 논문에서는 과거와 현재의 객체들을 각각 효율적으로 처리하기 위한 두 가지 히스토그램 CMH (Current Multidimensional Histogram)과 PMH(Past Multidimensional Histogram)을 제안한다.

시공간 객체들은 공간 속성과 유효시간을 가지며 이 객체들을 포함하는 히스토그램내의 버킷들은 시공간 객체들의 공간과 시간 속성들을 모두 포함할 수 있는 속성 값을 가진다. 하지만 현재 시간에 객체들의 분포를 균일하게 만들기 위한 버킷 분할 과정은 객체들의 유효시간의 끝이 결정되지 않은 상태이므로 시간 속성은 버킷 분할의 기준이 될 수가 없다. 따라서 현재 시간에 구축되는 히스토그램 CMH의 경우 시간 속성을 고려하는 것 자체가 무의미하다. 반면에 현재 시간에 히스토그램 CMH가 재구축 되면 이전 CMH가 포함하는 객체 정보들은 유효시간의 끝 시간을 가질 수 있게 된다. PMH는 이전 CMH를 기반으로 공간 속성과 시간 속성을 모두 고려하여 균일한 객체 분포를 가질 수 있도록 버킷 구조를 새로 구축한다. 그림 8은 전체적인 시스템의 구조를 나타내고 있다.

3.1 자주 사용하는 기호 정리

각 질의들은 점 질의(point queries) 뿐만 아니라 범위 질의(range queries)로 사용될 수 있다. 질의는 공간 영역과 유효시간을 가지고 있는데 유효 시간의 ls와 le가 같은 것을 시간적 점 질의 PQ(point queries with a timestamp)라고 하고 ls < le 인 조건을 만족하는 질의를 시간적 범위 질의 RQ(range queries with a time interval)라고 한다. 본 논문에서 사용하는 질의의 공간 영역은 사각형 형태로 주어지며 두 개의 점으로 2D 공간 영역을 표현한다.

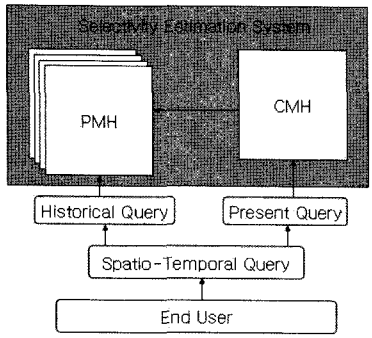


그림 8 선택도 추정 시스템

전체 작업 공간은 공간영역이 w×w 크기를 가지고 시간영역이 t크기를 가지는 셀들로 나누어지므로 한 히스토그램이 포함할 수 있는 전체 셀의 수는 w×w×t가 된다. 버킷은 이러한 셀의 시작과 끝 인덱스를 각각 유효시간 [ls, le]와 공간영역 [minX, minY, maxX, maxY]로 표현한다. *bi.ovr*은 b_i 버킷과 겹치는 객체들의 수를 저장하고, *bi.new*는 b_i 버킷에서 생성된 객체들의 수를 저장한다. 이 두 변수는 시간적 범위 질의에 의한 선택도 추정의 결과에서 이전 시점부터 살아온 객체들이 결과에 중복되는 것을 막아준다. 표 1은 본 논문에서 자주 사용하는 기호들을 정리하고 있다.

표 1 자주 사용하는 기호 정리

기호	설명
ls	유효시간의 시작 timestamp
le	유효시간의 끝 timestamp
minX	X축 최소 좌표
minY	Y축 최소 좌표
maxX	X축 최대 좌표
maxY	Y축 최대 좌표
B	히스토그램이 포함할 수 있는 최대 버킷 수
bi.ovr	i번째 버킷과 겹치는 객체들의 수
bi.new	i번째 버킷에서 생성된 객체들의 수
bi.cell	i번째 버킷이 포함하는 셀의 수
b_i*.dis*	i번째 버킷 ovr의 분산
$cell_i$*.t*	i번째 셀의 timestamp
w×w×t	히스토그램이 포함할 수 있는 전체 셀 영역

3.2 셀 초기화

CMH는 w×w×t 개의 전체 초기화된 셀들을 기반으로 B개의 버킷 집합을 생성한다. 시간의 변화에 따라 셀 정보는 객체의 정보 갱신에 따라 w×w개의 셀이 t개에 가까워진다. 예를 들어 CMH가 timestamp 10 동안 유지되었다면 이에 따른 셀의 수는 w×w×10개가 된다. 셀 집합에는 timestamp 10 동안 삽입, 삭제, 갱신된 모든 객체정보들을 포함하고 있다. 각각의 셀들은 해당 셀에서 생성된 객체수를 저장하는 변수 *new*와 셀에 겹치는 객체 수 *ovr*를 저장한다. 이 두 변수는 버킷을 구성할 때 근사 질의의 중복된 결과를 걸러주는 변수인 *b.ovr*과 *b.new*로 저장된다. 그림 9는 셀 초기화를 위한 간단한 예이다. 표현의 간편화를 위하여 공간 영역은 1차원으로 제한하였다. 그림 9(a)에서 각 세로선들의 양 꼭지점은 객체의 생성 시간과 종료 시간을 나타내고 회색 영역은 $cell_{11}$을 나타낸다. $cell_{11}$에서 시작되는 객체 수는 1개이고 겹치는 객체 수는 2개 이므로 $cell_{11}$*.new* = 1, $cell_{11}$*.ovr* = 2로 초기화가 된다. 이러한 셀 초기화는 매 timestamp마다 객체의 생성과 종료가 있는 셀에서 발생하게 된다. 그림 9(b)에 셀 초기화의 최종결과가 나타나 있다. 객체의 생성과 종료와 관계없는 셀들은 셀 초기화 작업을 하지 않기 때문에 초기화 작업 비용은 객체들의 갱신 횟수에 선형적이다. 여기에서 다음 정의 1에 따라 셀 초기화 과정을 최적화 시킨다.

정의 1. $cell_j$.ovr = $cell_i$.ovr + $cell_j$.new − $cell_i$.del. 단 $cell_i$는 $cell_j$와 공간 정보는 동일하고 바로 이전 timestamp를 가지며 $cell_i$.del는 $cell_i$에서 삭제된 객체 수이다.

예를 들어, 그림 9에 있는 회색 영역 $cell_{11}$의 ovr은 3인 것을 알 수 있다. 이 때 $cell_{11}$의 ovr은 정의 1에 의해 이전 시간의 셀의 정보를 이용하여 계산 되어 질 수 있고 그 값은 다음과 같다. $cell_{11}$.ovr = $cell_7$.ovr + $cell_{11}$.new − $cell_7$.del = 2 + 1 − 0 = 3.

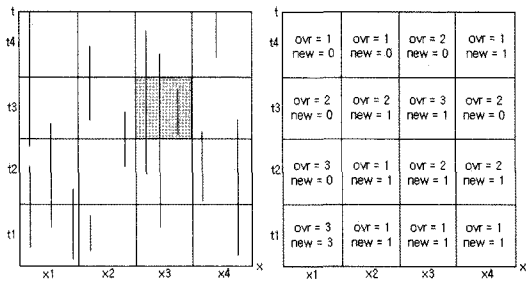


그림 9 셀 초기화: (a) 객체 정보; (b) 셀 정보

셀 초기화 과정에서 시간이 $cell_i$에 와 있고 이 때 $cell_j$가 초기화 될 때 정의 1에 따라 $cell_i$.ovr − $cell_i$.del의 값을 미리 미래 시간인 $cell_j$.ovr로 넣어두고 시간이 $cell_j$에 왔을 때 해당 영역에서 새로 추가된 객체 수($cell_j$.new)만 미리 저장된 $cell_j$.ovr에 더하면 $cell_j$에 대한 셀 초기화가 완료된다. 즉 특정 시간에 셀 초기화를 하는 경우에 객체의 생성과 삭제에 의해 객체 수가 변경되는 경우에만 해당 셀의 접근이 일어난다.

4. 현재 다차원 히스토그램

PQ를 위한 CMH는 전체 객체들을 요약하는 버킷들에 의해 구축된다. 버킷의 분할은 객체 분포를 균일하게 만들기 위해 이루어진다. 현재 시간에서 w×w 정규 셀을 가지는 초기 버킷으로부터 시작하여 B개만큼의 버킷이 되도록 이진 분할 처리(Binary Split Processing) 과정을 거친다. 이 때 각 버킷 분할의 과정은 여러 분할 경우 중에 원본 버킷 bi의 객체 분산 $b_i.dis$(예 : $b_i.dis = \sum_{k=1}^{b1.cell}(FAvg_i - c_k.ovr) / b_i.cell$)와 분할된 버킷 b_{i+1}, b_{i+2}의 평균 객체 분포 Avg($b_{i+1}.dis$, $b_{i+2}.dis$)의 차이가 가장 큰 경우를 기준으로 버킷 b_i를 분할한다. 각 버킷 분할 가치(Bucket Split Value) BSV의 계산 과정은 식 (6)에 나타나 있다.

$$b_i.wDis = b_i.dis * b_i.cellNum$$
$$BSV = b_i.wDis - Avg(b_{i+1}.wDis, b_{i+2}.wDis) \quad (6)$$

4.1 CMH 구축

CMH는 초기 셀 정보를 기반으로 BSV가 가장 큰 분할 기준으로 버킷을 이분 분할한다. 분할된 버킷은 이진 분할 트리(Binary Search Tree)에서 부모 노드가 되고 분할을 통해 생성된 두 버킷은 자식 노드가 된다. 이후의 버킷 분할을 위한 BSV 계산은 분할된 버킷의 정보만 변경되므로 새로 생성된 두 버킷의 BSV만 계산하면 된다. 버킷 분할 과정은 BST트리의 리프 노드(leaf node)의 총수가 B개가 되거나 모든 BSV 값이 음수가 될 때 종료한다. 예를 들어 그림 10은 현재 시간 now에서의 2차원 공간 셀 정보를 나타내며 그림 11은 그림 10을 기반으로 버킷 분할의 최종 결과를 나타낸다.

그림 12는 CMH를 구축하는 알고리즘이다. 첫 번째 단계에서 버킷의 수를 1로 설정하고 2 단계부터 5단계까지 첫 버킷의 영역을 전체 도메인으로 설정한다. 5단계부터 16단계까지는 버킷을 분할하는 과정을 보인다. 버킷 분할은 버킷의 수가 주어진 B개가 될 때까지 반복되며 매 분할마다 하나의 버킷을 두 개의 버킷으로 분할한다. 이 때 분할되는 버킷을 선정하는 기준은 가장 큰 BSV를 가지는 버킷을 선택하고 이 과정은 9단계에 나타난다. 만약에 모든 버킷의 BSV가 음의 값을 가진다면 더 이상 분할을 할 필요가 없어지므로 알고리즘을 종료하게 된다. 만약에 양의 BSV가 존재한다면 가장 큰 BSV를 가지는 버킷이 선택되고 12단계부터 15단계까지 버킷을 분할하는 과정을 거친다. 따라서 CMH 구축 알고리즘은 분할한 버킷의 수가 B개가 되거나 계산된 모든 BSV가 음의 값을 가지게 될 때 종료한다.

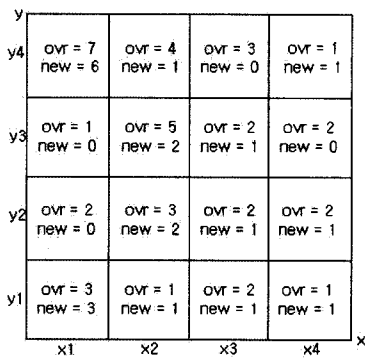


그림 10 timestamp now에서 셀 초기화 정보

4.2 CMH 버킷 갱신

CMH가 구축되고 시간이 지나 새로운 객체들의 생성과 이미 존재하는 객체들이 사라지는 경우 이에 대한 정보를 CMH내의 해당 버킷에 반영해야할 필요가 있다. 객체의 생성과 삭제 이벤트가 발생한 위치를 포함하는 버킷은 기존의 객체들을 요약하는 정보를 저장하고 새로운 버킷 정보를 생성한다. 새로운 버킷 정보는 버킷

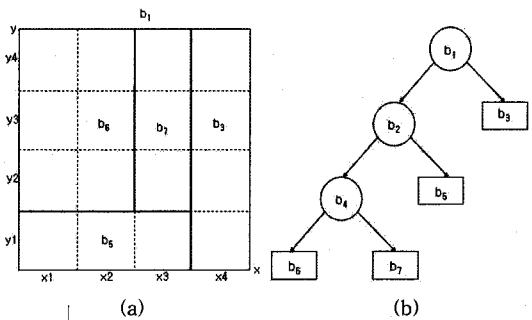


그림 11 CMH에서 버킷 분할의 최종 결과: (a) 분할된 버킷 정보; (b) 이진 분할 트리(BST)

```
Algorithm for constructing CMH
Begin
   bucketNum = 1;
1  b1.minX = 1; b1.minY = 1;
2  b1.maxX = w; b1.maxY = w;
3  b1.ls = now; b1.le = now;
4  while(bucketNum <= B)
5     for(int i=bucketNum; i!=0
6        && i>bucketNum - 1; i--)
7        calculate all BSV in bi
8           according to each axis x, y;
9     end for
10    select = bucket having maximum BSV;
11    if (select == NULL)
12       break;
13    split bselect into bbucketNum+1, bbucketNum+2
14    bbucketNum.lp = bbucketNum+1
15    bbucketNum.rp = bbucketNum+2
16    bucketNum++;
   end while
End
```

그림 12 CMH 구축 알고리즘

갱신이 발생한 시점을 ls로 가지고 now 시간을 le로 가지게 된다. 기존의 버킷 정보는 새로운 버킷 정보가 저장된 위치의 주소를 가지고 있어서 선택도 추정을 할 때 BST 트리에 의해 버킷을 검색한 뒤 질의의 시간에 맞는 버킷 정보를 저장된 주소를 따라 검색하면 된다. 예를 들어 timestamp 1에서 그림 10의 $cell_{12}$.ovr가 5로 변경되고 $cell_4$.ovr가 3으로 변경되면 버킷 집합에서 b_3이 두 셀을 포함하므로 버킷 갱신이 일어난다. 표 2는 timestamp 1에서 버킷 b_3의 갱신이 일어난 최종결과이다.

표 2 버킷 갱신

주소	버킷 이름	버킷 영역	ovr	new	dis	next	lifespan
1	b3	[x4, x4] [y1, y4]	6	3	0.25	5	[0, 1)
2	b5	[x1, x3] [y1, y1]	6	5	0.666667	null	[0, now)
3	b6	[x1, x2] [y2, y4]	22	11	3.888889	null	[0, now)
4	b7	[x3, x3] [y2, y4]	6	4	0.222222	null	[0, now)
5	b3	[x4, x4] [y1, y4]	11	8	2.1875	null	[1, now)

지금까지의 버킷 갱신 과정에서 각 timestamp마다 모든 버킷의 갱신 여부를 검사한다면 버킷 갱신을 위한 비용이 많을 수 있다. 따라서 객체 수가 변경된 버킷들만 검색하고 갱신하는 최적화 알고리즘이 필요하다.

최적화를 위하여 셀 초기화 과정에서 불 변수를 하나 두고 객체의 삽입과 삭제가 발생한 셀의 불 변수 값을 true로 변경한다. 버킷 갱신 과정은 불 값인 true인 셀을 포함하는 버킷이 있다면 갱신하고 해당 버킷이 포함하는 모든 cell의 불 변수 값을 false로 바꾼다. 그림 13은 갱신된 버킷만 검색하는 버킷 갱신 알고리즘이다. 이 알고리즘의 복잡도는 셀의 수를 n이라고 할 때 O(n)이다.

```
Algorithm for updating buckets
begin
1 For each cell_i overlapped with timestamp now inCMH
2    if (cell_i is changed)
3       update bucket containing cell_i;
4       set boolean variable of all cells
5          in the bucket to true;
   end if
6 end for
end
```

그림 13 bucketUpdate 알고리즘

5. 과거 다차원 히스토그램

시간이 지남에 따라 CMH에 있는 객체 분포들은 히스토그램 생성시의 균일한 성질을 잃어버리게 된다. 이때 객체 분포의 분산이 주어진 임계치를 넘어설 경우 객체 분포의 균일 성질을 유지하기 위해 히스토그램을 재구축한다. CMH 재구축은 객체들의 유효시간을 히스토그램 재구축 시점을 기준으로 now 시점을 포함하는 그룹과 그렇지 못한 그룹으로 분할한다. 예를 들어 객체 o_i의 유효시간이 [0, now)이고 히스토그램 재구축 시간이 8이라면 o_i는 유효시간 [0, 8)을 가지는 그룹과 유효시간 [8, now)를 가지는 그룹으로 분할된다. 이 두 객체 집합 중에 *now* 시점을 포함하는 객체 집합은 CMH의 재구축을 위하여 사용이 되고 *now* 시점을 포함하지 않는 객체 집합은 PMH 구축을 위해 사용된다. CMH 재구축 과정은 초기 구축과정과 같이 객체들의 공간 정보만을 고려하지만 PMH 구축은 CMH 재구축에 의해 PMH에서 사용할 객체 집합들의 유효시간의 끝이 재구축 시간으로 기록이 되어 있기 때문에 시간과 공간 정보들을 모두 사용하여 공간에서의 균일한 객체 분포뿐만 아니라 시간상으로도 균일한 객체 분포를 만들 수 있다. 5.1절에서는 이러한 PMH의 구축 방법을 기술한다.

5.1 PMH 구축

CMH 재구축에 의해 분리된 객체 집합은 CMH가 유지되는 동안에 변화된 객체 정보를 기록한 3차원 셀 배열에 저장되어 있다. PMH 구축은 이 3차원 셀 배열에 저장되어 있는 정보를 기반으로 이루어진다. PMH 구축도 CMH 구축과 동일한 과정을 거치지만 셀의 시간정보까지 포함하는 것이 다르다. 따라서 PMH 구축에 의해 생성되는 버킷들은 공간 정보와 시간 정보를 모두 가지게 되고 버킷 분할 과정에서 시공간 정보를 모두 이용한다. PMH에서의 버킷 정보는 다음 정의 2에 의해 new와 ovr을 결정한다.

정의 2. PMH에서의 버킷들은 해당 버킷이 포함하고 있는 셀 정보에 기반하여 버킷 매개변수 new와 ovr을 결정한다. 이 때 버킷 bi.new $= \sum_{cell_j \subseteq b_i} cell_j.w$ 이고 bi.ovr = bi.new + $cell_k$.ovr − $cell_k$.new, 단 $cell_k$.t = min($cell_j$.t), $cell_j \subseteq b_i$ 이다.

예를 들어, 그림 9(a)의 셀 정보를 기반으로 분할된 버킷 정보가 그림 14와 같다면 b_5.new = $cell_8$.new + $cell_{12}$.new + $cell_{16}$.new = 1 + 0 + 1 = 2가 되고 b_5.ovr = $cell_8$.ovr + $cell_{12}$.new + $cell_{16}$.new = 2 + 2 − 1 = 3이 된다. 이 때 그림 14의 버킷 b_5의 new와 ovr이 각각 2와 3임을 알 수 있다.

정의 2에 의해 버킷들은 셀과 거의 비슷한 형태인 자료구조를 가진다. CMH와 마찬가지로 PMH의 버킷 분할은 버킷의 수가 주어진 B 개 만큼 되거나 버킷 분할이 더 나은 객체 분포를 만들지 못할 때 까지 계속된다.

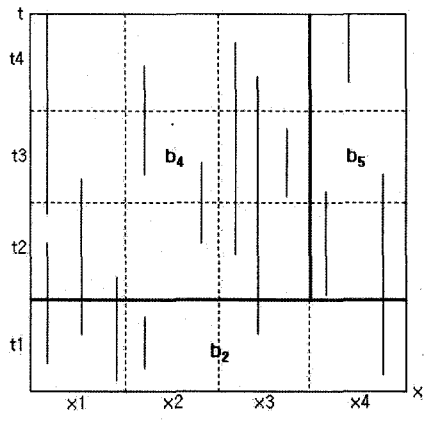


그림 14 PMH에서 분할된 버킷 정보

```
Algorithm for constructing PMH
begin
1  bucketNum = 1;
2  b1.minX=1; b1.minY=1; b1.maxX=w; b1.maxY=w;
3  b1.ls = cell1.t; b1.le = current timestamp;
4  PMHlast.ls = cell1.t; PMHlast.le = current timestan
5  while(bucketNum <= B)
6     for(i=bucketNum; i!=0 && i>bucketNum-1; i
7        calculate all BSV in bi
               according to each axis x, y, t;
8     end for
9     select = bucket having maximum BSV;
10    if(select == NULL)
11       break;
12    split bselect into bbucketNum+1, bbucketNum+2;
13    initialize information of bbucketNum+1 and bbucketNum+2
14       by theorem 2;
15    bbucketNum.lp = bbucketNum+1;
16    bbucketNum.rp = bbucketNum+2;
17    bucketNum++;
18 end while
   last++;
end
```

그림 15 PMH 구축 알고리즘

PMH가 포함하는 객체 수는 보통 CMH보다 많기 때문에 B가 CMH보다 많게 주어진다. 최근에 구축된 PMH는 CMH와 함께 주기억장치에 유지되고 PMH를 구축하기 위해 사용하였던 셀 배열의 유효시간을 PMH의 유효시간으로 할당한다.

그림 15는 PMH의 구축 알고리즘이다. 4단계에서 히스토그램에 유효시간을 할당, 7단계에서 시간 축 t가 추가, 13단계에서 정의 1에 근거한 버킷의 객체 정보 할당, 마지막으로 18단계에서 최근의 PMH를 나타내는 last 변수 값을 하나 증가 시키는 것 외에는 CMH 구축과 동일한 형태이다.

6. 선택도 추정

시공간 질의에 대한 선택도 추정은 구축된 PMH들과 CMH를 이용한다. 질의는 질의되는 시간의 유효시간의 시작과 끝이 같은지 아닌지에 따라 시간적 점 질의 PQ와 범위 질의 RQ로 나뉜다. 선택도 추정을 위해 선택되는 히스토그램은 두 질의 집합 PQ와 RQ의 유효시간에 따라 해당되는 PMH 또는 CMH를 선택한다. 6.1절에서는 시간적 점 질의 PQ에 대한 PMH와 CMH에서의 선택도 추정 기법을 설명하고 6.2절에서는 시간적 범위 질의 RQ에 대한 선택도 추정 기법을 기술한다.

6.1 시간적 점 질의

CMH와 PMH의 버킷 구조가 다르기 때문에 시공간 데이타의 선택도 추정은 PQ의 유효시간을 포함하는 히스토그램이 어떤 히스토그램이냐에 따라 달라진다. 만약 PQ의 유효시간과 겹치는 히스토그램이 CMH라면 먼저 PQ의 공간정보와 겹치는 CMH내의 버킷들을 모두 검색한다. 이 때 질의 공간 영역과 겹치는 각 버킷의 공간 영역을 해당되는 버킷의 전체 공간 영역으로 나누어 질의와 겹치는 공간 영역의 전체 공간 영역에 대한 비율을 구하고 이 비율에 버킷이 질의 시간에 가지는 객체수를 곱함으로써 질의와 겹쳐지는 버킷내의 객체수를 추정할 수 있다. 마지막으로 각각 구한 추정된 선택도를 모두 더함으로써 CMH에서의 PQ에 대한 전체 선택도 추정을 구할 수 있다. 다음 식 (7)과 (8)은 이러한 과정을 수식으로 나타낸 것이다.

$$Sel_j = b_j.ovr * OverlapArea(b_j)/area(b_j) \quad (7)$$

$$Sel = \sum_{j=0}^{k} Sel_j \quad (8)$$

$$Sel_j = b_j.ovr * OverlapVolume(b_j) \,/\, volume(b_j) \quad (9)$$

그림 16은 CMH에서 시간적 점 질의를 위한 선택도 추정 알고리즘이다. 1단계에서 PQ와 겹치는 모든 버킷을 검색한 후 3-4단계에서 질의 시간과 겹치는 버킷 정보를 검색한다. 6단계에서 검색된 각 버킷의 선택도 추정을 하고 7단계에서 모두 더하여 전체 선택도 추정을 구한다.

PMH에서의 시간적 점 질의를 위한 선택도 추정은 먼저 저장된 PMH 집합 중에서 질의 시간과 겹치는 PMH를 먼저 찾고 질의 시간과 공간 영역 모두 겹치는 버킷들을 검색한다. 각 버킷들에 대해 식 (9)를 적용하여 선택도 추정을 한 후 CMH와 마찬가지로 식 (8)을 적용하여 전체 선택도 추정을 구한다. PMH에서는 CMH와는 달리 선택도 추정을 할 때 질의의 시간과 공간 정보와 겹치는 버킷의 부피를 이용하여 계산한다. 그림 17은 PMH에서 시간적 점 질의를 위한 선택도 추정 알고리즘이다.

```
Algorithm for estimating selectivity in CMH
begin
1 Find buckets overlapped with PQ
2 For j=0 to the number of buckets
3   overlapped with query
4   while(bj.lifespan is not overlapped with PQ.lifespan)
5     bj = bj.next;
6   end while
7   Selj = bj.num * OverlapArea(bj) / area(bj);
8   Sel = Sel + Selj;
  end for
end
```

그림 16 CMH에서 PQ를 위한 선택도 추정 알고리즘

```
Algorithm for estimating selectivity in PMH
begin
1   Find buckets overlapped with PQ
2   For j=0 to the number of buckets
3     overlapped with query
4     Selj = bj.num *
5       OverlapVolume(bj) / volume(bj);
6     Sel = Sel + Selj;
    end for
end
```

그림 17 PMH에서 PQ를 위한 선택도 추정 알고리즘

6.2 시간적 범위 질의

PMH와 CMH를 이용한 시간적 범위 질의 RQ의 선택도 추정은 PQ의 선택도 추정을 확장한다. RQ를 위한 선택도 추정은 PQ와는 달리 질의가 범위 시간을 가지고 있기 때문에 여러 히스토그램이 선택도 추정을 위해 사용될 수 있다. 보통 히스토그램에서는 시간적으로 하나의 객체가 여러 버킷에 걸쳐 있을 수 있기 때문에 선택도 추정의 결과에 객체가 중복되어 적용되는 것을 방지 해야한다. 이를 위한 해결책으로 히스토그램 내의 각 버킷에 할당된 new와 ovr 변수를 이용하여 중복된 객체 적용을 피할 수 있다. 다음 정의 3에 의해 PMH에서의 선택도 추정 결과에 대해 객체가 중복 포함되는 것을 방지해 준다.

정의 3. 어떤 객체 집합들과 시간적 범위 질의 RQ가 버킷 bi에서 시작하여 bj까지 겹쳐져 있을 때 질의와 겹치는 버킷들의 객체 수는 bi.ovr + bi+1.new + ... + bj.new이다.

예를 들면, 그림 5.1의 초기화된 버킷 정보에서 질의 영역이 전체 영역이라고 할 때 질의와 겹치는 객체의 수는 정의 3에 의해 qr = b2.ovr + b4.new + b5.new = 6 + 5 + 2 = 13 개 이다. 이 때 질의가 전체 영역이므로 질의 결과는 모든 객체 수와 같고 그림 5.1에 나타나 있는 객체 수는 13이다.

CMH에서도 마찬가지로 버킷의 구조가 PMH의 간소화된 형태이므로 위의 정의 3을 이용하여 중복된 객체 처리를 방지 할 수 있다. 하지만 표 4.1에서와 같이 CMH의 경우 시간적으로 상위 버킷에 대한 주소를 하위 버킷들이 가지고 있으므로 위의 정의3을 적용하기가 매우 쉬우나 PMH에서는 시간적으로 하위 버킷이 질의와 겹친다 하더라도 상위 버킷 또한 질의와 겹친다는 보장이 없다. 예를들어 그림 6.3에서 질의 RQ는 b_2와 겹치고 있다. 이 때 b_2의 상위 버킷인 b_7은 RQ와 겹치지만 b_5와는 겹치지 않는다. 따라서 질의 RQ와 겹치는 버킷 집합에서 i) RQ.ls ≥ b_i.ls인 경우와 ii) RQ.ls < b_i.ls인 경우에 대해 검토해 보아야 한다. 그림 5.1에서 i)의 경우에 해당하는 버킷은 b_2이고 ii)의 경우에 해당하는 버킷은 b_7과 b_6이다. 따라서 버킷 b_i가 i)에 해당하는 버킷인 경우에 선택도 추정에 사용되는 객체 수 변수는 b_i.ovr이고 ii)에 해당하는 경우는 b_i.new이다.

RQ에 대한 선택도 추정은 CMH와 PMH에서 각각 식 (7)과 식 (9)에 정의 3을 적용시켜 사용한다. 다시 말해 그림 18의 b_7에 대한 선택도 추정은 식 (10)과 같이 계산되어지고 b_2에 대한 선택도 추정은 식 (9)를 이용하여 구할 수 있다. 각 버킷에 대한 선택도 추정을 하였으면 식 (8)을 이용하여 전체 선택도 추정을 구한다.

$$Sel_j = b_7.\text{new} * \mathit{OverlapVolume}(b_7) \;/\; \text{volume}(b_7) \tag{10}$$

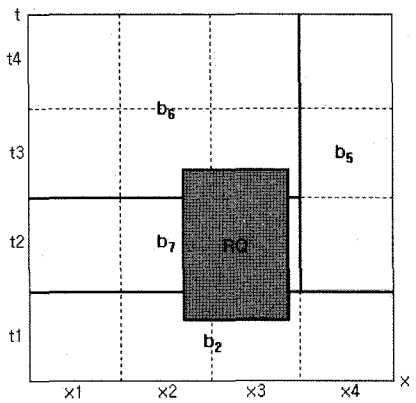


그림 18 PMH에서의 범위 질의

7. 실험 결과 및 분석

이번 절에서는 본 논문에서 제시한 CMH와 PMH 히스토그램의 성능 평가를 위한 실험 환경을 소개하고 이론적인 실험 모델로서 오류율 계산법과 실제 실험결과를 검토한다. 본 실험은 Intel Pentium4 northwood 2.8GHz CPU, 1GB RAM, 160GB HDD의 Windows XP 환경에서 수행하였으며 히스토그램의 구현을 위한 언어로 J2SE Development Kit 1.5를 사용하였고 개발

툴은 eclipse 3.1을 이용하였다. 히스토그램의 생성을 위해 이력 정보를 가지는 시공간 점 객체 1,000,000개를 무작위로 생성하여 히스토그램 구축에 이용하였다. 이렇게 생성된 객체 분포는 비교적 균일한 객체 분포를 가지게 된다. 표 3은 시공간 객체가 생성된 전체 시공간 도메인이다. 편중된 객체 분포를 만들기 위해 100,000개의 데이타를 시간: 0~299, 공간 Y: 0~2999, 공간 X: 0~2999에 추가 분포시키고 시간: 300~599, 공간 Y: 3000~5999, 공간 X: 3000~5999에 100,000 개의 데이타를 분포시켰다. 실험은 균일한 객체 분포와 편중된 객체 분포에 대해 각각 진행하였다.

표 3 시공간 객체가 존재하는 전체 도메인

객체 생성 변수	도메인 값
공간 도메인	10000
시간 도메인	800
객체의 유효시간의 최대 크기	300

7.1 이론적인 실험 모델

실험을 위한 상대 오류율 계산법은 식 (11)과 같으며 *Sel*은 히스토그램을 이용하여 질의 결과를 추정한 결과 값이고 *Sel'*는 실제 질의 결과 값이다.

$$Err = |(Sel - Sel')| / Sel', \text{ 단 } Sel'>0 \quad (11)$$

이 때 어떠한 질의의 실제 결과 Sel'이 0이면 오류율 측정이 불가능하게 되므로 1로써 대체하여 사전에 예외 상황에 대한 처리를 한다. 다수의 질의에 대한 오류율을 구한 뒤 이들의 평균을 구하기 때문에 강제로 0인 값을 1로써 대처한다고 하여도 질의의 수가 많으면 많을수록 실험 결과에 영향을 미치는 정도는 매우 작다. 그리고 특정 질의에 대한 편중된 결과를 해결하기 위해 Q_n개의 다수 질의에 대한 선택도 추정 오류율을 측정하고 그것의 평균을 구함으로써 실험에 보다 높은 신뢰도를 가지도록 한다. 따라서 최종 오류율은 식 (12)와 같다.

$$Avg(Err) = (\sum_{i=1\sim N} Err_i) / N \quad (12)$$

시간적 범위 질의와 점 질의에 대하여 비교할 연구가 없으므로 다음 4가지 실험 항목으로 각 데이타 집합에 대해 실험하였다.

(1) 고정된 임계치와 고정된 버킷을 가지면서 변화하는 질의 크기에 대한 추정 오류율
(2) 고정된 임계치와 고정된 질의크기를 가지면서 변화하는 버킷 수에 대한 추정 오류율
(3) 고정된 버킷 수와 질의 크기를 가지면서 변화하는 임계치에 대한 추정 오류율
(4) 고정된 버킷 수와 질의 크기를 가지면서 변화하는 임계치에 대한 히스토그램 재구축 횟수

7.2 질의 크기에 따른 추정 오류율 평가

표 4 실험 평가 항목(1)을 위한 고정 변수

고정 변수	변수 값
CMH에서의 버킷의 수	100
PMH에서의 버킷의 수	100
임계치	30%

표 5 질의 집합

Query set	Lifespan Size	Spatial Size
Query 1	30	500
Query 2	60	1000
Query 3	90	1500
Query 4	120	2000
Query 5	150	2500
Query 6	180	3000
Query 7	210	3500
Query 8	240	4000
Query 9	270	4500

첫 번째 실험 평가 항목으로 버킷 수와 임계치는 고정시키고 질의 크기를 변화시키며 선택도 추정 오류율을 측정하였다. 표 4는 CMH와 PMH 히스토그램이 가지는 버킷 수와 임계치를 나타내고 있다. 표 5는 질의 크기에 따른 9 가지 질의 집합을 나타낸다.

균일한 데이타 분포와 편중된 데이타 분포에 대한 최종 실험 결과는 그림 19와 그림 20에 각각 나타나 있다. 질의 크기가 커질수록 버킷 자체를 포함하는 경우가 늘어나므로 질의와 겹치는 각 버킷에 대한 오류율이 줄어든다. 따라서 전체 선택도 추정 결과에서 오류율이 질의 크기가 커질수록 점점 줄어든다. 하지만 질의 크기가 일정 크기를 넘어서면 버킷을 포함하는 경우 외에도 겹치는 수도 증가하므로 약간의 오류율이 증가한다.

이러한 현상은 객체의 분포가 균일한 경우와 편중된 경우 모두 나타나며, 질의 크기가 그 이상 커지면 다시

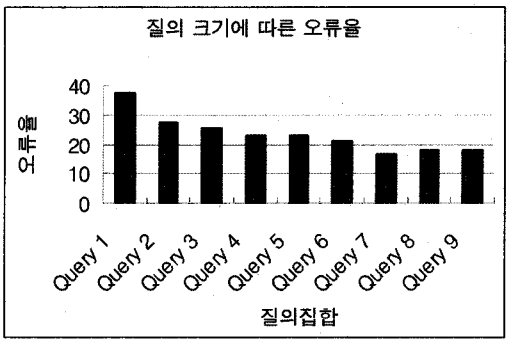


그림 19 균일한 객체 분포에 대한 질의 크기에 따른 오류율 변화

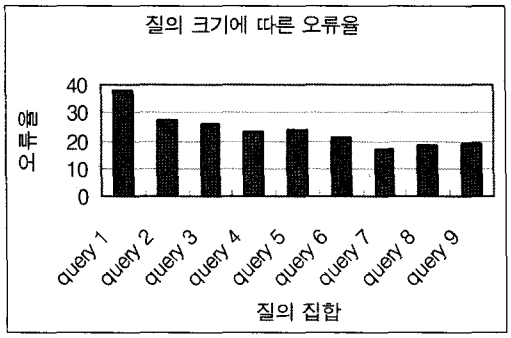


그림 20 편중된 객체 분포에 대한 질의 크기에 따른 오류율 변화

오류율이 줄어드는 경향을 보인다. 따라서 질의 크기에 따른 선택도 추정 오류율은 객체의 분포에 상관없이 질의 크기가 커질수록 감소하는 경향을 가진다.

7.3 변화하는 버킷 수에 대한 추정 오류율 평가

실험 평가 항목 (2)에서 버킷 수의 변화에 따른 선택도 추정 오류율을 구하기 위해 임계치와 질의 크기를 고정시켰다. 표 6은 히스토그램 구축을 위해 사용한 임계치와 선택도 추정을 위해 사용한 질의 영역을 나타내고 있다.

표 6 실험 평가 항목(2)를 위한 고정 변수

고정 변수	변수 값
임계치	30%
질의의 공간 영역 크기	3000
질의의 시간 영역 크기	180
질의의 수	100

그림 21은 균일한 데이타 분포에 대한 버킷 수에 따른 오류율이다. 히스토그램의 구축에서 버킷 분할이 버킷 내의 객체의 분포가 균일함 가정을 만족하게 하는 것인데 이미 전체 객체 분포가 균일하기 때문에 버킷의 수가 늘어나도 오류율의 변화는 거의 없다. 그림 22는 편중된 객체 분포에 대한 버킷 수에 따른 오류율이다.

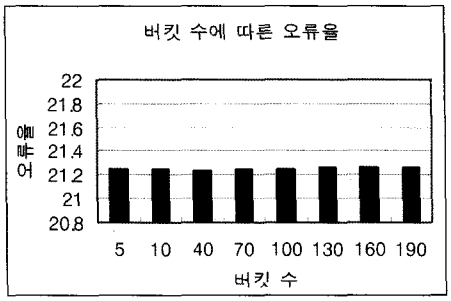


그림 21 균일한 객체 분포에 대한 버킷 수에 따른 오류율 변화

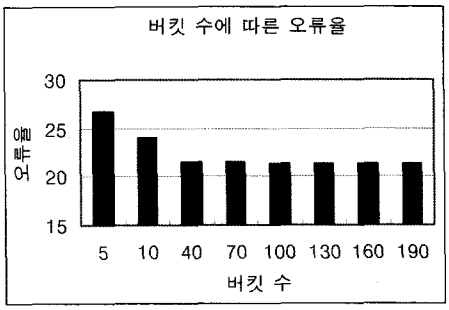


그림 22 편중된 객체 분포에 대한 버킷 수에 따른 오류율 변화

객체의 분포가 편중되어 있기 때문에 버킷의 수가 작으면 버킷 내의 균일함 가정을 만족시킬 만큼 버킷을 분할하지 못하므로 여전히 편중된 버킷이 존재할 수 있다. 따라서 적당한 수 이상의 버킷이 되어야 선택도 추정 오류율이 안정됨을 알 수 있다. 본 논문의 실험에서는 그림 22에서 보는 것과 같이 버킷의 수가 약 40개 이상이 되어야 추정 오류율이 안정되었다. 이 실험의 결과로 객체 분포의 편중 현상이 심할 경우 버킷 수에 따른 추정 오류율의 변화도 커진다는 것을 알 수 있다.

7.4 변화하는 임계치에 대한 추정 오류율 평가

실험 평가 항목 (3)에서 임계치의 변화에 따른 선택도 추정 오류율을 구하기 위해 버킷의 수와 질의 크기를 고정시켰다. 표 7은 히스토그램 구축을 위해 사용한 버킷의 수와 선택도 추정을 위해 사용한 질의 영역을 나타내고 있다.

표 7 실험 평가 항목(3)을 위한 고정 변수

고정 변수	변수 값
CMH에서의 버킷의 수	100
PMH에서의 버킷의 수	100
질의의 공간 영역 크기	3000
질의의 시간 영역 크기	180
질의의 수	100

그림 23은 균일한 객체 분포에 대한 임계치에 따른 오류율 변화를 보이고 있다. 실험 평가 항목(2)의 결과와 마찬가지로 균일한 객체 분포에서는 임계치에 변화에 따른 선택도 추정 오류율의 변화는 거의 없다. 시간의 변화에 따라 초기 CMH에 존재하는 각 버킷내의 객체 분포가 균일 가정을 만족 못하는 경우가 생기는데 이를 해결하기 위해 CMH를 재구축하고 PMH를 생성한다.

이 때 CMH의 재구축은 객체 변화량이 주어진 임계치를 초과할 경우 발생된다. 하지만 균일한 객체 분포의 경우 시간이 지나더라도 삽입되는 객체들의 분포도 균

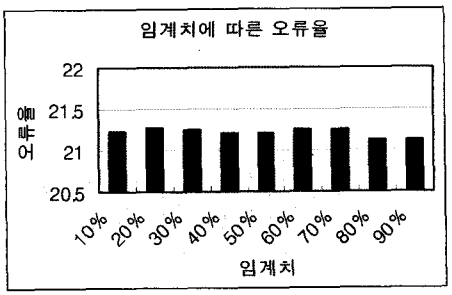


그림 23 균일한 객체 분포에 대한 임계치에 따른 오류율 변화

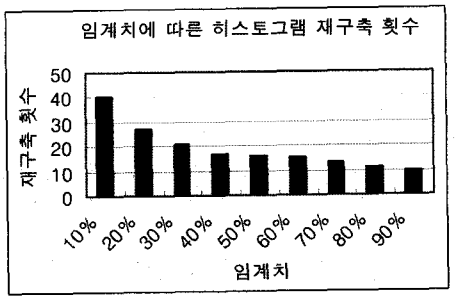


그림 25 균일한 객체 분포에 대한 임계치에 따른 히스토그램 재구축 횟수

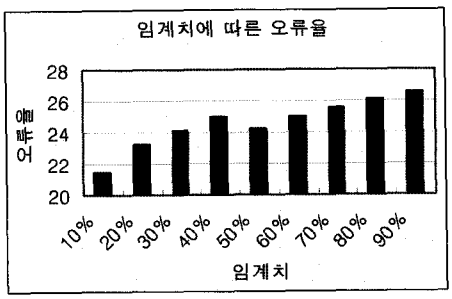


그림 24 편중된 객체 분포에 대한 임계치에 따른 오류율 변화

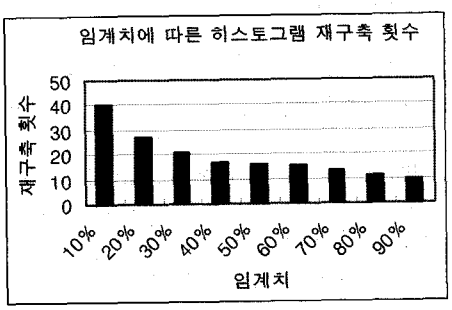


그림 26 편중된 객체 분포에 대한 임계치에 따른 히스토그램 재구축 횟수

일하기 때문에 히스토그램을 재구축 하지 않아도 여전히 CMH에서의 버킷 내의 객체 분포가 균일할 수 있다. 따라서 그림 23과 같이 임계치가 높아져 CMH의 재구축이 발생하지 않아도 선택도 추정 오류율의 변화가 거의 없다. 반면에 편중된 객체 분포의 경우 시간이 지남에 따라 CMH 내의 객체 분포가 점점 편중되므로 재구축을 하지 않을 경우 선택도 추정 오류율이 증가할 수 있다. 따라서 그림 24와 같이 임계치가 증가하여 CMH의 재구축이 늦어질수록 오류율은 증가한다.

7.5 변화하는 임계치에 대한 히스토그램 재구축 횟수 평가

실험 평가 항목 (4)에서 임계치의 변화에 따른 히스토그램 재구축 횟수를 구하기 위해 버킷의 수와 질의 크기를 고정시켰다. 이 때 사용한 버킷 수와 질의 크기는 표 7.5의 내용을 사용하였다.

그림 25와 그림 26은 변화하는 임계치에 따른 히스토그램 재구축 횟수를 측정한 결과를 나타내며 객체 분포에 상관없이 임계치가 증가함에 따라 히스토그램 재구축 횟수가 작게 발생하는 것을 알 수 있다. 실험 평가 항목 (4)와 실험 평가 항목 (3)의 결과에 따라 적당한 수준의 임계치를 선택하는 것이 매우 중요하다는 것을 알 수 있다.

8. 결 론

본 논문에서는 시공간 데이타베이스에서 시퀀스 데이타를 위한 선택도 추정 기법으로 히스토그램 CMH와 PMH를 제안하였다. 제안된 히스토그램은 최초로 과거와 현재까지의 시퀀스 데이타에 대한 선택도 추정이 가능하다. 또한 시공간 영역 내에서 시퀀스 데이타의 시간상 범위 질의가 가능하고 현재와 과거를 담당하는 두 히스토그램간의 유기적인 히스토그램 유지 기법으로 질의에 대한 효과적인 선택도 추정을 할 수 있다. 이 때 히스토그램의 유지 기법으로 CMH의 객체 편중도가 주어진 임계치를 넘어서는 경우 CMH를 재구축하는 임계치 기법을 사용하였다. 제안된 히스토그램의 재구축은 공간 정보와 시간정보를 모두 고려한 PMH를 구축하고 현재 시간에서의 객체들을 기반으로 하는 새로운 CMH를 구축하여 객체 편중도를 균일하게 함과 동시에 히스토그램의 저장 공간을 효율적으로 관리할 수 있다.

시험 결과로서 균일한 객체 분포에 대한 실험 항목(2)와 (3)에서 버킷의 수나, 임계치의 변화와 상관없이 선택도 추정 오류율이 낮음을 알 수 있다. 반면에 편중된 객체 분포에 대해서는 임계치와 버킷 수의 변화가 일정 수 이상 되기 전까지는 오류율이 높아짐을 알 수 있다. 이러한 결과에서 CMH와 PMH의 히스토그램 재구축과 버킷 분할 과정이 편중된 객체 분포를 균일하게 하는데

효과적임을 알 수 있다. 실험 항목(1)의 결과에 대해 질의 크기가 커질수록 버킷 자체를 포함하는 경우가 늘어나므로 질의와 겹치는 각 버킷에 대한 오류율이 줄어든다. 따라서 객체 분포에 상관없이 전체 선택도 추정 결과에서 오류율은 질의 크기가 커질수록 점점 줄어든다. 하지만 질의 크기가 일정 크기 이상이 되면 버킷을 포함하는 경우 외에도 겹치는 수도 증가 하므로 오류율이 약간 증가한다. 마지막으로 임계치가 높을 경우 히스토그램의 재구축이 작게 발생하므로 임계치와 히스토그램 재구축은 반비례하다. 따라서 실험 항목(3)과 연관하여 최적의 임계치를 찾는 것이 선택도 추정 오류율을 유지하는데 중요하다.

앞으로의 연구 과제는 PMH에서 히스토그램 버킷 분할 과정 시 공간과 시간 정보 모두에 효과적인 버킷 분할 알고리즘을 개발하여 선택도 추정 오류율을 줄이는 방법을 고려할 것이다.

참 고 문 헌

[1] Acharya, S., Poosala, V., and Ramaswamy, S., "Selectivity Estimation in Spatial Databases," In ACM SIGMOD, USA, pages 13-24, 1999.

[2] Aboulnaga, A. and Naughton, J. "Accurate Estimation of the Cost of Spatial Selections," In ICDE, pages 123-134, 2000.

[3] Poosala V., Yannis E., Ioannidis, Peter J., Haas., and Eugene J. Shekita, "Improved Histograms for Selectivity Estimation of Range Predicates," In ACM SIGMOD, NY, USA, pages 294-305, 1996.

[4] Yossi Matias, Jeffrey Scott Vitter, and Min Wang, "Wavelet-Based Histogram for Selectivity Estimation," In Proceedings of ACM SIGMOD international conferences on Management of data, pages 448-459, 1998.

[5] J. S. Vitter, M. Wang, and B. Iyer, "Data cube approximation and histograms via wavelets," In Proceedings of Seventh International Conference on Information and Knowledge Management, pages 96-104, Washington D.C., November 1998.

[6] K. Charkrabarti, M. Garofalakis, R. Rastogi, and K. Shim, "Approximate Query Processing Using Wavelets," In Proc. of the 26th Intl. Conf. on Very Large Data Bases, September 2000.

[7] Wang, M., Vitter, J., S., Lim, L., and Pdmanabhan, S., "Wavelet-Based Cost Estimation for Spatial Queries," In The 7th International Sysposium on Spatial and Temporal Databases(SSTD), CA, USA, pages 175-196, July 2001.

[8] Matias, J.S. Vitter, and M. Wang, "Dynamic Maintenance of Wavelet-Based Histogram," In Proc. of the 26th INtl. Conf. on Very Large Data Bases, September 2000.

[9] M. Garofalakis and P.B. Gibbons, "Wavelet Synopses with Error Guarantees," Bell Labs Tech. Memorandum, December 2001.

[10] Antonios Deligiannakis and Nick Roussopoulos, "Extended Wavelets for Multiple Measures," In Processdings of the 2003 ACM SIGMOD International Conference on Management of Data.

[11] Choi, Y. and Chung, C., "Selectivity Estimation for Spatio-Temporal Queries to Moving Objects," In ACM SIGMOD, pages 440-451, 2002.

[12] Tao, Y., Sun, J., and Papadias, D., "Selectivity Estimation for Predictive Spatio-Temporal Queries," ICDE, pages 417-428, 2003.

[13] Hadjieleftheriou, M., Kollios, H., and Tsotras, V J., "Performance Evaluation of Spatio-temporal Selectivity Estimation Techniques," In The 15th Int. conference on Science and Statistical Database Management (SSDBM), pages 202-211, 2003.

[14] Zhan, Q. and Lin, X., "Clustering Moving Objects for Spatio-temporal Selectivity Estimation," In ADC, pages 123-130, 2004.

[15] Sun, J., Papadias, D., Tao, Y., and Liu, B,. "Querying about the Past, the Present, and the Future in Spatio-Temporal Databases," In ICDE, pages 202-213, Mar. 2004.

[16] Tao, Y., Papadias, D., and Sun, J., "The TPR*-tree: An Optimized Spatio- Temporal Access Method for Predictive Queries," In Proceedings of the 29th Very Large Data Bases Conference, Berlin, Germany, pages 790-801, 2003.

[17] Lee, J. and Shin, B., "Histogram-based Selectiviry Estimation in Spatio-Temporal Databases," In Jonual of Korea Information Processing Society, Vol. 12-D, No.1, Feb. 2005.

신 병 철

2004년 2월 충북대학교 컴퓨터교육과(이학사). 2006년 2월 충북대학교 대학원 컴퓨터교육과 석사. 2006년~현재 에어포인트(주) 기술연구소 연구원. 관심분야는 질의 최적화, 시공간 데이타베이스, GIS 등

이 종 연

1985년 2월 충북대학교 전자계산기공학과(공학사). 1987년 2월 충북대학교 대학원 전자계산기공학과(공학석사). 1999년 2월 충북대학교 대학원 전자계산학과(이학박사). 1989년 비트컴퓨터(주) 개발부. 1990년~1994년 현대전자산업(주) 소프트웨어연구소 주임연구원. 1994년~1996년 현대정보기술(주) CIM사업부 책임연구원. 1999년~2003년 삼척대학교 정보통신공학과 조교수. 2003년~현재 충북대학교 컴퓨터교육과 부교수. 관심분야는 질의 최적화, 시공간 데이타베이스, bioinformatics, ubiquitous computing, u-learning, RFID middleware, GIS.