# 시공간 데이터베이스를 위한 히스토그램 기반 선택도 추정 기법

이 종 연[†] · 신 병 철[††]


## 요 약

시공간 데이터베이스의 영역에는 크게 이동객체를 다루는 시계열 데이터베이스 영역과 이력객체를 다루는 서열 데이터베이스 영역으로 나뉜다. 최근에는 시공간 데이터베이스의 질의 최적화를 위한 선택도 추정 연구가 활발히 진행되었으나, 기존 연구는 주로 시계열 데이터베이스의 선택도 추정에 의한 질의 최적화에 중점을 두었고 서열 데이터베이스에 대한 질의 최적화 연구는 전무하였다. 따라서 본 논문에서는 시공간 데이터베이스의 질의 최적화를 위한 T Minskew 히스토그램을 구축하고 이를 이용한 선택도 추정 기법을 제안한다. 또한 임계치 기법을 이용한 효과적인 히스토그램 유지 기법을 제안한다.


# Histogram-based Selectivity Estimation Method in Spatio-Temporal Databases

Jong-Yun Lee[†] · Byoung-Cheol Shin[††]


## ABSTRACT

The processing domains of spatio-temporal databases are divided into time series databases for moving objects and sequence databases for discrete historical objects. Recently the selectivity estimation techniques for query optimization in spatio temporal databases have been studied, but focused on query optimization in time series databases. There was no previous work on the selectivity estimation techniques for sequence databases as well. Therefore, we construct T-Minskew histogram for query optimization in sequence databases and propose a selectivity estimation method using the T Minskew histogram. Furthermore we propose an effective histogram maintenance technique for good performance of the histogram.


**키워드 : 시공간 데이터베이스(Spatio-temporal Databases), 질의 최적화(Query Optimization), 선택도 추정(Selectivity Estimation), 히스토그램(Histogram)**

## 1. 서 론

최근 시공간 데이터에 대한 관심이 늘어나면서 이를 사용하는 여러 애플리케이션이 개발되었고 시공간 DBMS의 필요성이 대두되었다. 이러한 시공간 DBMS는 시간에 따라 변화하는 공간객체들의 변화를 효과적으로 관리할 수 있어야 한다. 시공간 DBMS의 영역에는 크게 이동 공간객체에 대한 부분(time-series data)과 이력 공간객체에 대한 부분(sequence data)이 있다. 이동 객체에 대한 질의는 현재의 객체 정보에 기반하여 미래의 어느 시점에서 이 객체의 공간 정보가 어떻게 변화하는 것인가를 예측하는 것이다. 예를 들면, "Find all moving objects that overlap with query area A during 30 minutes."와 같은 질의를 들 수 있다. 이력 공간 질의는 과거의 어떤 시점에서 질의의 공간 영역과 겹치는 객체들을 찾아내는 것이다. 예를 들면, "Find all objects that were overlapped with query area A at time t."와 같은 질의를 들 수 있다.

시공간 데이터베이스에 대한 연구는 지금까지 많이 진행되어 왔다. 이를 위해 기존의 공간 데이터베이스나 시간 데이터베이스에서 두 분야를 결합시켜 시공간 데이터베이스에 대한 연구 분야로 접근하였고 그에 대한 결과로 [1]이나 [2]와 같은 색인이나 질의 처리에 관한 연구가 활발하게 진행 되었다. 이러한 시공간 데이터베이스에서 질의 최적화를 위해 중요하게 다루는 선택도 추정은 기존의 공간 선택도 추정의 연구 [3-7]에서 확장되어 왔다고 볼 수 있다.

기존의 시공간 데이터베이스 선택도 추정 [8-11]은 이동 객체에 대한 연구가 주로 이루어져 왔지만 본 논문에서는 Minskew 히스토그램[3]을 확장하여 이력 공간객체를 기반한 선택도 추정이 가능한 T-Minskew 히스토그램의 구축 방법과 효과적인 히스토그램의 유지기법을 제안한다. 본 논문에서 제안하는 T-Minskew 히스토그램의 특징은 다음과 같다. (i) Minskew 히스토그램을 timestamp 별로 구축하고 유지한다. (ii) 너무 많은 히스토그램 재구축을 방지하기 위해 히스토그램 임계치를 두어 재구축 횟수를 줄이면서 만족스러운 선택도 추정률을 유지하도록 한다. 본 논문에서

※ 이 논문은 2004년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음.
† 정 회 원 : 충북대학교 컴퓨터교육과 교수
†† 준 회 원 : 충북대학교 대학원 컴퓨터교육과
논문접수 : 2004년 11월 18일, 심사완료 : 2004년 12월 15일

제안하는 히스토그램은 기본적으로 2차원 객체의 특성 시점에 대한 질의를 주로 다룬다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 3장에서는 본 논문에서 제안하는 T-Minskew 히스토그램 구축과 이를 이용한 선택도 추정 기법을 기술한다. 4장에서는 제안한 기술을 위한 실험 평가를 하고 5장에서 결론을 정리한다.

## 2. 관련 연구

### 2.1 기존의 시공간 선택도 추정

[8]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 차원별로 구해진 선택도를 곱하여 2차원 공간에서의 선택도 추정을 한다. 따라서 [8]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 $Sel$은 [9]에서 다음과 같이 제시하고 있다. 이 때 $q_R$은 사각 질의를 나타내고 2차원 공간에서 각 축별로 하한과 상한 값을 가진다.

$$Sel = \frac{VT}{U^2}[(q_{R.xmin} - q_{R.xmin}) + (q_{R.ymin} - q_{R.ymin})] + \frac{(q_{R.xmin} - q_{R.xmin})(q_{R.ymin} - q_{R.ymin})}{U^2} \quad (1)$$

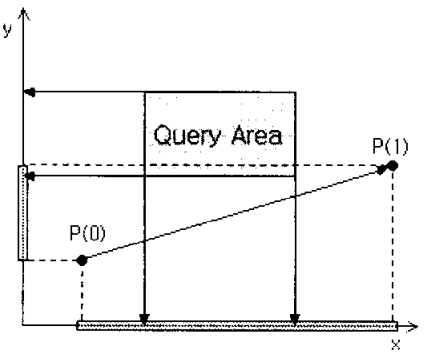
이 때 [8]에서 제시한 방법으로 선택도 추정을 한다면 각 차원 별로 수식 (2)와 같은 선택도 추정이 가능하고 이를 곱하면 수식 (3)과 같아지므로 선택도가 원래보다 높게 추정이 됨을 알 수 있다. 수식 (2)에서 $q_{Ri}$는 질의의 한 차원을 가리키며 상한과 하한을 가지고 있다.

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

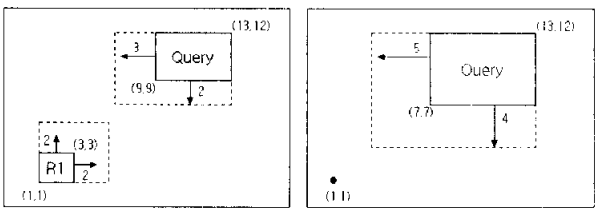
$$Seli = \frac{VT}{2U}[(q_{R.xmax} - q_{R.xmax}) + (q_{R.ymax} - q_{R.xmax} + \frac{(q_{R.xmax} - q_{R.xmax})(q_{R.ymax} - q_{R.ymax})}{U^2} + \frac{V^2}{4U^2} \quad (3)$$

(그림 2.1)은 [9]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.

[9]에서는 [8]에서 제시한 이동 객체 표현을 2차원 공간으로 확장시켰다. 그리고 [8]에서의 초과 선택도 추정에 대하여 2차원 공간을 1차원 공간으로 사상시키는 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 [9]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 2.2)와 같이 객체들을 간소화하는 기술을 사용하여 풀어내고 있다. 히스토그램의 재구축은 전체 데이터집합에서의 갱신율이 정해진 임계치를 초과할 경우에만 일어나므로 재구축 빈도가 [8]에 비하여 줄어들었다.




(그림 2.1) 2차원 공간에서 초과된 추정

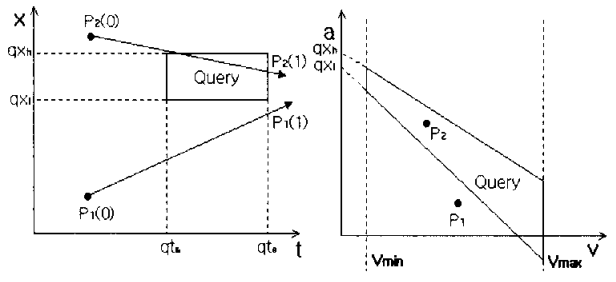



(a) 이동 객체와 질의; (b) 이동 객체의 간소화

(그림 2.2) 이동 객체의 간소화 과정

[10]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 2.3)은 공간-시간 그래프와 속도 절편 그

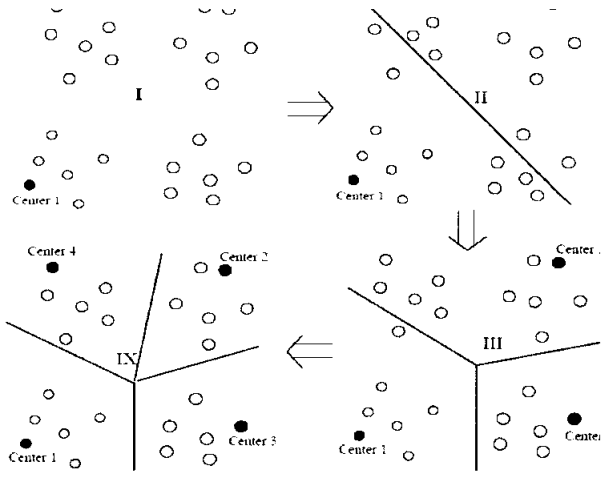
래프의 예를 보인다. [10]에서의 선택도 추정은 점 객체들이 존재하는 전체 공간을 질편-속도 그래프에서 MinSkew 알고리즘을 사용하여 버킷들로 분해한 뒤 질의 영역과 겹치는 버킷들의 영역을 계산함으로써 구할 수 있다.




(a) 공간-시간 그래프; (b) 속도-절편 그래프

( 그림 2.3 ) Hough 변환을 사용한 그래프 변환 :

[11]에서는 클러스터링 기술을 기반으로 한 버킷 분할 히스토그램을 제시하고 있다. 이전까지의 이동객체를 위한 선택도 추정의 연구에서 대부분 Minskew 히스토그램을 확장한데에 반하여 더 효율적인 공간 버킷 분할을 위해 클러스터링 기술을 이용했다. 이는 비슷한 성질을 가지는 객체는 가까운 거리에 존재한다는 것이다. 다시 말해 두 객체가 가깝다면 초기 위치와 속도, 객체 크기가 비슷할 수 있다. 이동 객체간의 거리 계산을 위해 유클리드 계산법을 확장하여 적용하였으며 기본적인 클러스터링 방법을 위해 (그림 2.4)와 같은 방법을 사용한다. 히스토그램의 정제를 위해 이미 구축된 히스토그램의 버킷에서 어떤 객체를 뽑아 다른 버킷에 넣어봄으로써 더 나은 히스토그램이 되게 하는 기법도 제안하고 있다.




( 그림 2.4 ) Gonzalez Clustering

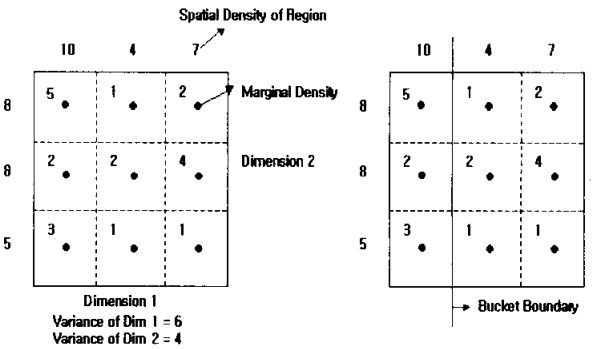
### 2.2 Minskew

공간 선택도 추정 기법은 [3, 5, 12-13]에서 많이 사용되었다. [3]에서는 공간 선택도 추정을 위한 히스토그램인 Minskew를 제안하고 있다. Minskew 히스토그램은 편중된 객체들의 버킷 분할을 통한 분배를 통하여 균일하게 만드는데 있다. 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. $C$는 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 수식 (4)과 같이 표현하며 전체 편중도의 가중치는 수식 (5)로 표현한다. 최종 분할 경우의 선택은 가중치가 가장 작은 것으로 한다.

$$B_i.skew = \frac{1}{|C|} \sum_{i=1}^{vmberofcellinBi} (C_i.den - Avg(den))^2 \quad (4)$$

$$Minskew = \sum_{i=1}^{n} (B_i.num \times B_i.skew) \quad (5)$$

(그림 2.5)는 Minskew 히스토그램에서 버킷 분할의 예를 보이고 있다. (그림 2.5) (a)를 보면 Dimension 1의 분산은 6이고 Dimension 2의 분산은 4임을 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.




(a) 원본 히스토그램; (b) 버킷 분할 기준

( 그림 2.5 ) Minskew 히스토그램의 공간 분할:

(그림 2.5) (b)는 선택된 분할 축을 기준으로 가능한 분할 경우의 수에서 수식 (4)을 이용하여 분할된 두 버킷의 편중도 skew를 구하고 수식 (5)를 이용하여 최종 가중치를 구한 뒤 가중치가 가장 작은 분할 선을 기준으로 공간을 두개의 버킷으로 분할한 것이다. 수식 (5)를 이용하여 각 분할 가능한 경

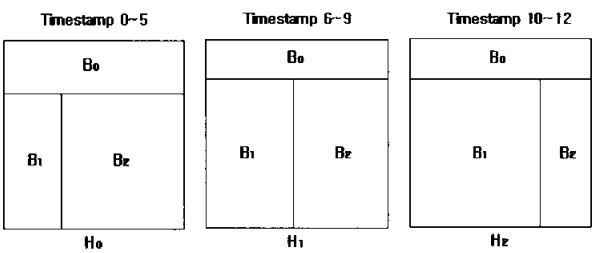
우의 가중치를 구하면 첫 번째 경우의 가중치는 28.14가 나오며 두 번째 경우는 37.38이 나온다. 따라서 첫 번째 분할이 편중도를 낮출 수 있는 방법이고 그 결과는 (그림 2.5) (b)와 같다.

## 3. T-Minskew 히스토그램

이 장에서는 Minskew 히스토그램[3]을 이용한 이력 공간 질의의 선택도 추정을 위한 T-Minskew 히스토그램을 제안한다. T-Minskew 히스토그램은 2차원 공간 구조에 시간을 고려한 히스토그램이다. 히스토그램내의 버킷은 공간 정보를 가지면서 히스토그램의 timestamp내에 존재한다는 것을 보장한다. 따라서 질의와 겹치는 버킷을 검색하기 위해서는 우선 질의의 timestamp와 겹치는 히스토그램을 검색하고 질의의 공간 영역과 겹치는 해당 히스토그램 내의 버킷을 검색하면 된다.

(그림 3.1)은 간단한 T-Minskew 히스토그램의 timestamp 0부터 12까지 형태의 예를 보이고 있다. *timestamp* 0에 히스토그램 $H_0$가 생성되어 5까지 지속되다가 timestamp 6에 새로운 히스토그램 $H_1$이 생성되고 *timestamp* 10에 히스토그램 재구축에 의해 다시 새로운 히스토그램 $H_2$가 생성되었음을 보이고 있다. 이때 각 히스토그램이 재구축 될 때에는 히스토그램 내의 객체 정보가 변화하여 버킷의 분할이 다른 형태로 이루어졌음을 알 수 있다.

본 논문에서는 기본적으로 특정 시점을 가지는 사각형 질의 $Qr$에 대한 선택도 추정을 하고 있다. 또 *timestamp*별로 히스토그램을 생성하므로 여러 개의 히스토그램이 존재하고 각 히스토그램은 유효 시간 $[t_{start}, t_{end})$를 가지고 있다. 각각의 히스토그램 내에는 여러 버킷이 존재하고 공간영역 정보와 버킷이 포함하는 객체수를 저장한다. 전체 선택도 $Sel$은 각각의 버킷에 대한 선택도 추정 $Sel_i$을 합하여 구한다. 다음 <표 3.1>은 본 논문에서 사용하고 있는 기호들에 대한 정리이다.




( 그림 3.1 ) T-Minskew 히스토그램

### 3.1 T-Minskew 히스토그램 구축

시공간 이력 객체의 선택도 추정을 위한 T-Minskew 히스토그램을 구축하기 위하여 공간 히스토그램 Minskew를 확장하여 다음 두 가지 가정을 세운다.

**[가정 1] 히스토그램 재구축** : 히스토그램 $H_i$의 시간 간격은 $H_i$가 생성 된 시점부터 i+1번째 재구축이 일어난 $timestamp_{now}$ - 1로 한다.

**[가정 2] 히스토그램 갱신** : 히스토그램이 재구축 되지 않을 경우 객체 변화를 해당 버킷에 적용하기 변화된 객체 수를 시간과 함께 버킷에 기록한다.

〈표 3.1〉 기호 테이블

| 기 호 | 의 미 |
|---|---|
| $Qr$ | 윈도우 질의 |
| $Qr.t$ | 윈도우 질의의 *timestamp* |
| $B_i.MBR = [B_i.x_{min}, B_i.y_{min}, B_i.x_{max}, B_i.y_{max}]$ | i번째 버킷의 2차원 공간 좌표 |
| $B_i.num$ | i번째 버킷이 포함하는 객체 수 |
| $H_i$ | i번째 히스토그램 |
| $H_i.t = [H_i.t_{start}, H_i.t_{end})$ | i번째 히스토그램의 유효시간 |
| $H_i.var$ | i번째 히스토그램의 객체 변화량 |
| $timestamp_{now}$ | 현재시간 |
| $Sel_i$ | i번째 버킷에 대한 선택도 |
| $Sel$ | 전체 선택도 |
| $OverlapArea(B_i.MBR)$ | i번째 버킷과 겹치는 질의 영역의 넓이 |
| $area(Bi.MBR)$ | i번째 버킷의 넓이 |
| $avg(den)$ | 셀의 평균 밀도den의 평균 |

히스토그램 $H_0$가 실제 객체를 기반으로 timestamp 0에 구축되었을 때 유효 시간은 [0, *NOW*)을 가진다. 시간이 지나 새로운 timestamp가 1이 되었을 때 재구축을 해야 되는 지에 대한 판단을 하게 되는데 객체의 변화율과 임계치를 비교함으로써 히스토그램 재구축 여부를 결정한다. 만약 timestamp 1에서의 객체 변화율이 임계치를 넘어서지 않았을 경우는 timestamp 0에 구축된 히스토그램 $H_0$는 timestamp 1에도 유지가 되며 유효시간은 여전히 [0, *now*)로써 변화가 없게 된다. 이 때 객체가 새로 생성이 되었을 경우에는 객체가 가지는 공간 좌표를 포함하는 버킷의 객체수를 하나 증가시키고 객체가 삭제된 경우나 이전하였을 경우는 객체를 포함하는 버킷의 객체수를 하나 감소시키고 이전한 위치의 버킷의 객체수를 하나 증가시킨다. 버킷의 정보가 변경되었을 경우에 변경된 시점과 보유하고 있는 객체수를 기록하여 객체 변화의 이력정보를 기록한다. 이러한 작업은 기존의 버킷이 가지는 객체 수는 사라지지 않고 변화된 객체 수만 시간과 함께 이력 정보로써 기록하여 선택도 추정을 할 때 변화된 객체수도 함께 검사하여 좀 더 정확한 선택도 추정을 가능하게 한다. 가정 1에 의해 timestamp 4에서 객체 변화율이 임계치를 넘어 섰을 경우 히스토그램의 유효시간 끝점을 timestamp 3으로 기록하여 유효시간을 [0, 3)으로 갱신하고 timestamp 4에는 이 시점의 데이터베이스에서 살아 있는 객체들을 기반으로 새로운 히스토그램 $H_1$를 생성한 뒤 유효시간을 [4, *now*)로 할당한다.

(그림 3.2)는 이러한 히스토그램 재구축 알고리즘이다. 단

계 1에서 이전 timestamp까지의 객체 변화량에 현재 timestamp의 변화량을 더한다. 단계 2에서는 변화량에 전체 객체 수를 나눈 값이 임계치를 넘어 섰는지를 검사한다. 만약 임계치를 넘어 선다면 단계 3에서 i번째 히스토그램의 유효 시간의 끝인 $H_{i.tend}$를 현재 timestamp의 바로 이전으로 기록하고 단계 4에서 새로운 히스토그램 $H_{i+1}$을 생성한다. 단계 5와 6에서 각각 새로운 히스토그램 $H_{i+1}$의 유효시간을 기록하고 알고리즘을 종료한다. 만약 히스토그램의 객체 변화율이 정해진 임계치보다 높지 않다면 변경된 객체 정보를 히스토그램에 갱신해야하므로 단계 8에서 updateBucket 프로시저를 호출한다. (그림 3.3)은 가정2에 따라 히스토그램의 갱신을 위한 updateBucket 프로시저 알고리즘의 기술이다.

**Algorithm rebuildHistogram**

```
1  Hi.var += calculate variation of objects
              in universal space during
              [timestamp_now - 1, timestamp_now);
2  if (ratio(Hi.var / N)) is over a given threshold
3     Hi.t_end = timestamp_now - 1;
4     Create new histogram H_i+1;
5     H_i+1.t_start = timestamp_now;
6     H_i+1.t_end = now;
7  else
8     call updateBucket procedure to apply
       changed information of objects at buckets;
9  end if
```

**End rebuildHistogram**

( 그림 3.2 ) 히스토그램 재구축 알고리즘

**Algorithm updateBucket**

```
1  Find bucket B_i overlapped with the space extent
   of objects;
2  Calcuate the number of objects changed within
   the bucket B_i;
3  if (B_i.num is changed)
4     Append changed B_i.num with
        timestamp t at the bucket B_i;
5  end if
```

**End updateBucket**

( 그림 3.3 ) 버킷 갱신 알고리즘

updateBucket 프로시저의 단계 1에서 변경된 객체의 공간 영역과 겹치는 버킷 Bi를 찾는다. 단계 2에서는 가정2에 따라 버킷 Bi에서 변경된 객체수를 계산하여 만약 변화가 있다면 단계 4에서 버킷 $B_i$에 timestamp t와 함께 변경된 객체 수를 추가한다.

히스토그램 재구축을 결정짓는 객체의 변화율은 히스토그램이 생성된 시점부터 실제 객체의 생성, 삭제, 갱신 횟수가 전체 객체 수에 비해 얼마나 일어났는지에 따른다. 예를 들어 전체 객체 수 N이 100이고 히스토그램 유지 기간 동안의 객체 변화 횟수가 20이라면 20%의 객체 변화율을 가지게 되며 만약 임계치가 15%라면 재구축이 일어나게 된다.

### 3.2 T-Minskew 히스토그램을 이용한 선택도 추정

시공간 이력 질의에 대한 선택도 추정을 위해서는 우선 질의의 timestamp와 겹치는 히스토그램을 찾은 뒤 히스토그램 내에서 질의의 공간 영역과 겹치는 버킷들을 찾는다. 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 가지는 객체수를 곱함으로써 질의와 겹쳐지는 버킷내의 객체수를 추정할 수 있다. 마지막으로 각 버킷에서 추정된 객체수를 더함으로써 전체 선택도를 추정할 수 있게 된다. 수식 (6)과 (7)는 이러한 과정을 수식으로 나타낸 것이다.

$$Sel_j = B_j.num * OverlapArea(B_j.MBR)/area(B_j.MBR) \quad (6)$$

$$Sel = \sum_{j=0}^{k} Sel_j \quad (7)$$

**Algorithm Selectivity**

```
1  Find a histogram H_i
     that satisfy (H_i-t ⊃ Qp-t or Qr-t),
2    where (0 ≤ i ≤ number of histogram);
   Find buckets that overlap with
3    query area within Hi
   For j=0 to the number of buckets overlapped with query
4    num = B_j.num overlapped with Qr-t;
5    Sel_j = num *
       OverlapArea(B_j-MBR) / area(B_j-MBR);
6  end for
7  For j=0 to the number of buckets
     overlapped with query
8    Sel = Sel + Sel_j;
9  end for
```

**End Selectivity**

( 그림 3.4 ) 선택도 추정 알고리즘

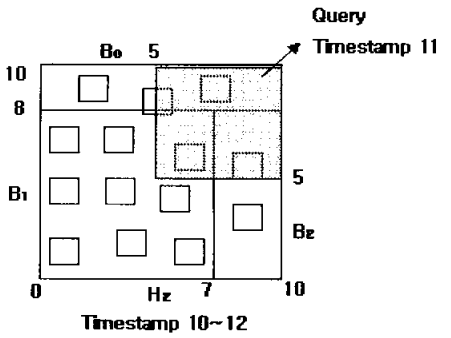
(그림 3.4)는 T-Minskew를 이용한 선택도 추정 방법의 알고리즘을 나타내고 있다. 단계 1에서 질의의 *timestamp*를 포함하는 히스토그램을 찾고, 단계 2에서는 검색된 히스토그램 내에서 질의의 공간 영역과 겹치는 버킷들을 찾는다. 단계 4에서 버킷 Bj내에서 객체의 수를 저장하는 하나 이상의 *Bj.num* 중에 질의의 시간과 겹치는 *Bj.num*을 임시변수 *num*에 저장하고, 단계 5에서 이 버킷의 선택도를 수식 (6)을 이용하여 추정한다. 단계 8에서 전체 선택도를 구하기 위해 각각 구해진 버킷의 선택도를 모두 합하는 것으로 알고리즘은 종료된다.

구체적인 예를 위해 (그림 3.1)에서 질의의 timestamp와 겹치는 히스토그램 $H_2$를 가져온 것을 (그림 3.5)에서 보인다. 질의 사각형의 공간 영역과 각 버킷의 정보는 다음과 같고 계산의 간편화를 위하여 질의의 시간과 모두 겹치는 것으로 가정한다.

$Qr.MBR$ = [5, 5, 10, 10]
$B_0.MBR$ = [0, 8, 10, 10], $B_0.num$ = 3
$B_1.MBR$ = [0, 0, 7, 8], $B_1.num$ = 9
$B_2.MBR$ = [7, 0, 10, 8], $B_2.num$ = 2

이 때 (그림 3.5)에서 각 버킷의 선택도를 추정과 전체 선택도 추정은 다음과 같이 계산된다.

$Sel_1$ = 3 * 10 / 20 = 1.5
$Sel_2$ = 9 * 6 / 56 = 0.96
$Sel_3$ = 2 * 9 / 24 = 0.75
$Sel = \sum_{j=0}^{3} Sel_j$ = 1.5 + 0.96 + 0.75 = 3.21




(그림 3.5) T-Minskew를 이용한 선택도 추정

## 4. 실험 평가

이번 절에서는 본 논문에서 제시한 T-Minskew 히스토그램의 성능 평가를 위한 실험 환경을 소개하고 이론적인 실험 모델로써 오류율 계산법과 실제 실험결과를 제시한다. 본 실험은 Intel Pentium4 northwood 2.8GHz CPU, 512MB RAM, 160GB HDD의 Windows XP 환경에서 수행하였으며 T-Minskew 히스토그램의 구현을 위해 Visual Studio 6.0을 사용하였다. T-Minskew 히스토그램의 생성을 위해 이력 정보를 가지는 공간 객체 10,000개를 무작위로 생성하여 히스토그램을 구축하였다.

### 4.1 이론적인 실험 모델

T-Minskew의 선택도 추정의 정확도를 판단하기 위해 여러 가지 실험 기준에 따른 상대적인 오류율을 측정하도록 한다. 실험을 위한 상대 오류율 계산법은 수식 (8)과 같다. $Sel$은 T-Minskew를 이용하여 질의 결과를 추정한 값이고 $Sel'$는 실제 질의 결과 값이다.

$$Err = |Sel - Sel'| / Sel', \text{ 단 } Sel' > 0 \quad (8)$$

이 때 어떠한 질의의 실제 결과가 0가 되는 경우 오류율 측정이 불가능하게 되므로 1로써 대체하여 사전에 예외 상황에 대한 대처를 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 $Q_n$개의 다수 질의에 대한 선택도 추정 오류율을 측정하고 그것의 평균을 구함으로써 실험에 보다 높은 신뢰도를 가지도록 한다. 따라서 최종 오류율은 수식 (9)과 같다.

$$Avg(Err) = (\sum_{i=1}^{Qn} Err_i) / Qn \quad (9)$$

만약 Minskew 히스토그램의 평균 오류율이 $M_{err}$이라면 T-Minskew의 오류율 $TM_{err}$은 $M_{err}$보다 α만큼 증가한다. 왜냐하면 임계치 기법에 의하여 히스토그램을 유지할 때 객체의 변화에 따른 히스토그램의 재구축을 하지 않고 이전 단계에서 사용했던 히스토그램을 그대로 쓰기 때문이다. 또한 추가된 오류율 α는 높은 임계치일 수록 히스토그램의 재구축률이 감소하므로 증가하게 된다. 따라서 Minskew 히스토그램에 대한 T-Minskew의 오류율 $TM_{err}$은 수식 (10)과 같다. 하지만 본 논문에서 제시한 updateBucket 알고리즘에서 보는 바와 같이 객체의 변화에 따른 버킷들의 정보 변화를 위해 히스토그램의 재구축을 통하지 않고 임계치 내에서 각각의 버킷에게 맡겨두기 때문에 임계치에 따른 히스토그램 재구축 횟수는 감소하고 오류율의 변화도 거의 없다.

$$TM_{err} = M_{err} + \alpha \quad (10)$$

### 4.2 실험 결과 분석

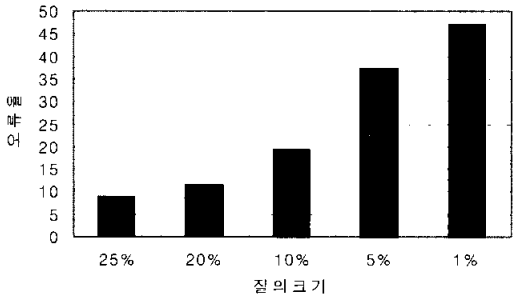
실제 실험은 다음의 변수에 따라 수행하여 T-Minskew 히스토그램에 대한 평가를 하도록 한다.

(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 고정된 질의 크기를 가지면서 변화하는 임계치에 대한 히스토그램 재구축 횟수
(4) 고정된 버킷 수와 질의 크기를 가지면서 변화하는 임계치에 대한 추정 오류율

#### 4.2.1 변화하는 질의 크기에 대한 추정 오류율 평가

질의 크기에 대한 추정 오류율의 변화를 보기 위해 임계치를 30%로 고정하고 히스토그램의 버킷 수를 50으로 고정 한 뒤 각각 질의 크기에 따라 무작위로 만든 100개의 질의에 대해 실험하였다. 질의 크기는 각 축별로 전체 공간에 대한 비를 나타낸 것으로 만약 전체 공간의 x축 영역과 y축 영역이 1000이고 라고 하였을 때 질의 크기가 10%라면 각 축별로 질의 크기는 100의 크기를 가지게 되므로 2차원 공간이라면 전체 공간에 비해 1%의 크기를 가지게 됨을 알 수 있다. 실험의 결과 (그림 4.1)과 같이 질의 크기가 작으면 작을수록 오류율이 증가하였다. 이러

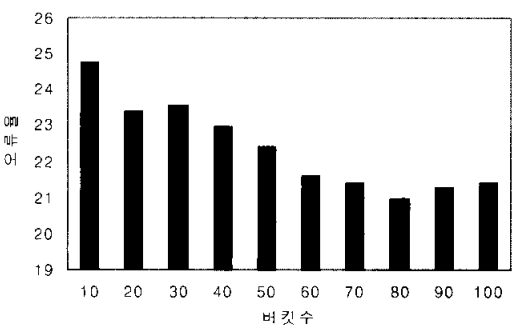
한 실험결과를 가지는 이유는 질의 크기가 작으면 작을수록 추정된 결과값과 실제 질의 결과값이 차이가 많지 않음에도 불구하고 상대적인 오류율 계산법에 따라 오류율은 상대적으로 많아질 수 있기 때문이다. 예를 들어 실제 질의 결과가 1이고 추정된 결과가 1.6이면 60%의 오류율을 가지는 반면 실제 질의 결과가 10이고 추정된 질의 결과가 10.6이라면 단순히 6%의 오류율을 가진다. 따라서 실험에서 사용한 10,000개의 객체 수 보다 더 많은 객체에 대한 실험을 하면 질의 크기가 작아도 질의 결과가 많아지게 될 것이므로 이러한 현상은 줄어들 것이다.




( 그림 4.1 ) 질의 크기에 대한 추정 오류율

4.2.2 변화하는 버킷 수에 대한 추정 오류율 평가

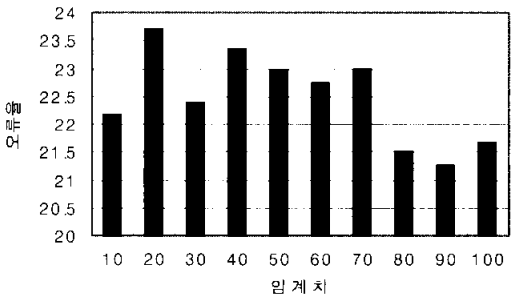
버킷 수에 따른 선택도 추정 오류율에 대한 평가를 위해 임계치를 30%로 고정하고 질의 크기는 10%로 고정한 뒤 무작위로 질의를 100개 생성하여 실험하였다. 버킷의 수가 작으면 작을수록 하나의 버킷이 포함하는 공간 영역 또한 증가 하고 최적의 편중도를 만족할 수 없기 때문에 오류율이 증가할 수 있다. 다시 말해 버킷의 크기가 크면 클수록 편중된 객체 분포를 버킷으로 분할 시켜 균일하게 바꿀 수가 있게 된다. 하지만 너무 많은 버킷의 수는 오히려 최적의 히스토그램 상태를 억지로 분할 할 수도 있기 때문에 오히려 오류율이 증가할 수도 있다. 이러한 실험의 결과가 (그림 4.2)에 나타나 있다. 버킷의 개수 80까지는 오류율이 하락하지만, 그 이후에는 오히려 오류율이 증가한다. 따라서 히스토그램의 상태에 따라 최적의 버킷 수를 결정하는 일은 선택도 추정 오류율을 낮추는 데에 있어 중요하다.




( 그림 4.2 ) 변화하는 버킷수에 따른 추정 오류율

4.2.3 변화하는 임계치에 대한 히스토그램 재구축 횟수 평가

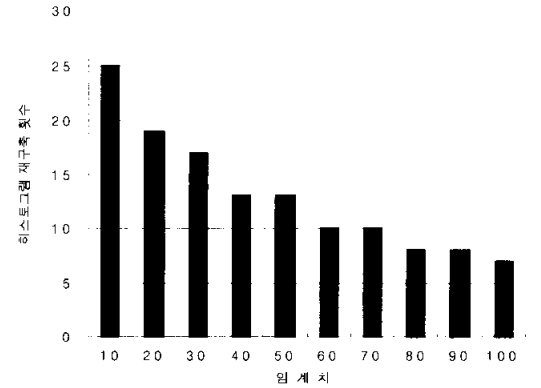
변화하는 임계치에 대한 히스토그램의 재구축 횟수를 평가하기 위해 버킷 수는 50개로 고정시키고 4.2.2절에서 사용하였던 10%의 크기를 가지는 100개의 질의를 사용하였다. 이 때 임계치가 크면 클수록 시간에 따른 객체의 변화량이 많아도 히스토그램을 유지하고 갱신하기 때문에 재구축 횟수가 줄어든다. (그림 4.3)은 임계치를 5부터 100까지 5단위로 변화시키며 재구축 횟수를 측정한 것이다. 실험 결과 역시 임계치의 증가에 따라 재구축 횟수가 점점 줄어들고 있다.




( 그림 4.3 ) 임계치 변화에 따른 히스토그램 재구축 횟수

4.2.4 변화하는 임계치에 대한 추정 오류율 평가

변화하는 임계치에 대한 선택도 추정 오류율 평가를 위해 버킷 수는 50개로 고정 시키고 4.2.2절에서 사용하였던 10%의 크기를 가지는 100개의 질의를 사용하였다. 이러한 실험의 결과가 (그림 4.4)에 나타나 있다. 4.1절과 같이 임계치가 변화하더라도 선택도 추정 오류율의 변화는 거의 없음을 알 수 있다. 이것은 히스토그램의 작은 재구축 횟수로도 낮은 오류율을 가지는 선택도 추정을 할 수 있음을 의미한다.




( 그림 4.4 ) 변화하는 임계치에 따른 추정 오류율

## 5. 결 론

본 논문에서는 이력 공간객체의 선택도 추정을 위한 T-Minskew 히스토그램을 제안하였다. 히스토그램의 유지를 위하여 임계치 기법을 이용하여 객체의 변화율에 따른

재구축 판단을 하게 되므로 잦은 재구축을 방지했다. 이러한 임계치 기법은 객체의 변화율이 지정된 임계치를 넘어서기 전에는 변화한 객체 정보를 히스토그램 내의 버킷의 갱신을 통해 처리를 하는 것이다. 이 기법을 통하여 이전 시간의 객체수를 저장하고 변화된 객체수를 시간과 함께 저장함으로써 효율적인 히스토그램 유지를 할 수 있다. T-Minskew 히스토그램은 편중된 객체들을 균일한 구조로 바꾸기 위해 Minskew 히스토그램 기법을 사용하였다.

실험결과로써 질의 크기가 작을수록 선택도 추정 오류율이 증가함을 알 수 있었다. 그리고 히스토그램이 가질 수 있는 버킷의 개수가 많을수록 추정 오류율이 감소하였다. 그러나 최적의 버킷 개수를 넘어 서는 경우는 오히려 오류율이 증가했다. 임계치가 크면 클수록 히스토그램의 재구축 횟수는 줄어들었는데 이때 히스토그램의 재구축 횟수가 작더라도 특정 시점에서의 공간 선택도 추정의 오류율이 높아지지 않았다. 이러한 실험 결과는 T-Minskew 히스토그램이 효과적인 히스토그램 유지 기법을 통해 재구축 횟수는 줄이면서 선택도 추정 오류율을 유지할 수 있기 때문이다.

앞으로의 연구 과제는 시공간 범위 질의의 선택도 추정이 가능한 히스토그램을 구축하고 유지하는 기술을 개발하는 것이다.

## 참 고 문 헌


[1] 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, pp.790-801, 2003.

[2] Tao, Y. and Papadias, D., "Time- Parameterized Queries in Spatio-Temporal Databases," In Proceedings of ACM SIGMOD international conferences on Management of Data, pp.334-345, 2002.

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

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

[5] 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, pp.294-305, 1996.

[6] 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, pp.175-196, July 2001.

[7] Nikos Mamoulis and Dimitris Papadias, "Selectivity Estimation of Complex Spatial Queries," In the 7th International Sysposium on Spatial and Temporal Databases(SSTD), CA, USA, pp.156-174, July, 2001.

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

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

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

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

[12] 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, pp.448-459, 1998.

[13] Lee, J., Kim, D., and Chung, C., "Multi-dimensional selectivity estimation using compressed histogram information," In Proceeding of ACM SIGMOD international conferences on Management of Data, pp.205-214, 1999.



### 이 종 연

e-mail : jongyun@chungbuk.ac.kr
1985년 충북대학교 전자계산기공학과 (공학사)
1987년 충북대학교 대학원 전자계산기공학과 (공학석사)
1999년 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 조교수
관심분야 : 질의 최적화, 시공간 데이터베이스, 바이오 데이터 마이닝, Intelligent GIS, CIM 등

### 신 병 철

e-mail : suemirr@nate.com
2004년 충북대학교 컴퓨터교육과(이학사)
2004년~현재 충북대학교 대학원 컴퓨터교육과 석사과정
관심분야 : 질의 최적화, 시공간 데이터베이스, GIS 등