

누적밀도 웨이블릿 히스토그램을 이용한 공간 선택을 추정

(Spatial Selectivity Estimation using Cumulative Wavelet Histograms)

지정희[†] 정재혁^{**} 류근호^{***}
 (Jeong Hee Chi) (Jae Hyuk Jeong) (Keun Ho Ryu)

요약 선택을 추정 기법들의 공통적인 목표는 데이터의 요약 정보를 적은 저장 공간에 유지하고, 추정된 값과 질의 결과에 대한 오차를 최소화하는 것이다. 방대한 양의 공간 데이터에 대한 선택을 추정의 경우, 정확한 결과를 얻기 위해서는 공간 데이터의 분포를 반영하는 요약 정보를 필요로 하며, 그러한 요약 정보를 생성하기 위해서는 많은 저장 공간을 필요로 한다.

따라서, 이 논문에서는 적은 저장 공간을 사용하면서, 정확성 높은 선택을 추정하는 누적밀도 웨이블릿 히스토그램을 제안한다. 이 히스토그램은 기존의 누적밀도 히스토그램에 유지되는 각 서브 히스토그램에 대해 웨이블릿 변환을 적용함으로써, 보다 적은 저장 공간에서 높은 정확도의 선택을 얻을 수 있다. 우리는 실험결과를 통하여 기존 히스토그램의 저장 공간에 25%~50%만을 사용하여 높은 정확도의 선택을 추정 결과를 보임으로써, 기존의 다른 선택을 추정 기법들이 갖는 저장공간에 대한 제약사항을 해결함과 동시에 적은 저장공간을 사용하여 높은 정확도의 선택을 추정이 가능함을 확인 하였다. 이 논문에서 제안된 기법은 공간 데이터베이스에서의 공간 범위 질의에 대한 정확성 높은 선택을 추정하기 위해 사용될 수 있다.

키워드 : 질의 처리, 선택을 추정, 공간 히스토그램, 웨이블릿

Abstract The purpose of selectivity estimation is to maintain the summary data in a very small memory space and to minimize the error of estimated value and query result. In case of estimating selectivity for large spatial data, the existing works need summary information which reflect spatial data distribution well to get the exact result for query. In order to get such summary information, they require a much memory space.

Therefore in this paper, we propose a new technique cumulative density wavelet Histogram, called CDW Histogram, which gets a high accurate selectivity in small memory space. The proposed method is to utilize the sub-histograms created by CD histogram. The each sub-histograms are used to generate the wavelet summary information by applying the wavelet transform. This fact gives us good selectivity even if the memory size is very small. The experimental results show that the proposed method simultaneously takes full advantage of their strong points - gets a good selectivity using the previous histogram in (25%~50%) memory space and is superior to the existing selectivity estimation techniques. The proposed technique can be used to accurately quantify the selectivity of the spatial range query in databases which have very restrictive memory.

Key words : query processing, selectivity estimation, spatial histogram, wavelet

· 이 연구는 유비쿼터스 바이오 정보 기술 연구 센터와 ETRI의 연구비 지원으로 수행되었음

† 학생회원 : 충북대학교 전자계산학과
 jhchi@dblab.chungbuk.ac.kr

** 정 회 원 : 한국에너지기술연구원
 roknavy7@dblab.chungbuk.ac.kr

*** 종신회원 : 충북대학교 전자계산학과 교수
 khryu@dblab.chungbuk.ac.kr

논문접수 : 2005년 1월 20일

심사완료 : 2005년 6월 23일

1. 서 론

공간 데이터베이스 시스템은 일반적으로 점(point), 선(line), 면(surface), 다각형(polygons)과 같은 공간 객체를 저장하고 관리한다. 최근 이러한 공간 데이터의 크기가 급격히 증가하면서 공간 질의 처리의 중요성이 더욱 높아지고 있다. 일반적으로 공간 질의 처리는 객체간

의 위상관계를 이용한 공간 질의가 주어지게 되면, 실제 데이터 분포를 최대한 근사화 할 수 있는 요약정보(summary data)를 기반으로 실제 질의 결과에 대한 근사 값을 추정하여, 최적의 질의 수행계획을 선택함으로써 처리된다. 따라서, 질의 결과의 크기를 정확하게 추정하는 것은 질의 처리 성능에 있어서 매우 중요한 요소이며, 이에 대한 많은 연구가 이루어지고 있다.

기존의 데이터베이스 시스템에서의 선택을 추정은 실제 데이터 분포를 근사하는 요약정보를 구성하고, 이를 이용하여 질의 처리를 수행하는데 초점을 맞추고 있다. 예를 들면, 질의 최적화기는 질의에 대해 요약정보를 기반으로 수행된 선택을 추정결과를 사용하여, 가능한 질의 실행계획의 비용을 계산하고, 가장 비용이 적은 최적의 실행계획을 선택하고 있다. 또한, 대규모 데이터에 대한 분석을 수행하는 데이터 웨어하우스에서는 빠른 근사 응답을 위한 OLAP 질의에 대해서도 선택을 추정이 사용되고 있다. 이러한 기존 데이터베이스 시스템에서의 선택을 추정 기법은 매개 변수적 기법, 샘플링 기법, 히스토그램 기법 등이 사용되어 왔다[1]. 이 기법들 중, 히스토그램 기법은 매우 적은 공간을 사용하고, 사전 데이터 분포에 대한 정보를 사용하지 않음으로, 상용 데이터베이스 시스템에서 많이 사용되고 있다.

공간 데이터베이스 시스템에서는 기존의 데이터베이스 시스템에서와 같이 실제 데이터 분포를 근사하는 요약 정보를 구성하고, 이를 기반으로 선택을 추정하고 있다. 그러나 공간 데이터는 다양한 영역과 크기를 가지며, 여러 형태로 분포하고 있다. 최근에는 이러한 공간 데이터에 특성을 고려한 다양한 히스토그램 기법들[2-4]이 제안되었다. 여러 히스토그램 기법들 중, MinSkew 히스토그램[2]은 영역 객체의 편중도에 따른 선택율의 오차를 최소화하기 위해 제안되었다. 또한, 영역 객체가 여러 버킷에 걸쳐 있을 경우 발생하는 중복계산 되는 문제를 해결하기 위해 누적밀도(cumulative density) 히스토그램[3]과 오일러(euler) 히스토그램[4]이 제안되었다. 그러나, 이러한 기법들은 충분한 저장 공간을 갖는다는 가정을 기반으로 수행된다. 따라서, 기존의 기법들은 공간 범위가 증가 하거나 요약 정보에 대해 제한된 저장 공간이 사용될 경우, 타당한 결과를 얻기 어렵다. 이러한 이유는 일반적으로 요약 정보를 저장하기 위한 저장 공간은 제한된 반면, 공간 데이터의 양은 증가하여, 많은 양의 공간 데이터에 대한 정보를 적은 양의 저장 공간을 갖는 요약 정보에 반영하여 선택율을 추정함으로써 많은 오차를 발생 할 수 있기 때문이다.

따라서, 이 논문에서는 공간 객체 분포의 특성을 충분히 고려함과 동시에 이에 대한 공간 요약정보의 손실을 최소화하여, 제한된 저장 공간에서의 타당한 공간 선택

율을 추정하기 위한, 새로운 공간 선택을 추정기법으로 누적밀도 웨이블릿(CDW : Cumulative Density Wavelet) 히스토그램을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 공간 히스토그램과 이 논문의 기반이 되는 웨이블릿 변환에 대해 살펴보고, 3장에서는 이 논문에서 사용되는 용어에 대한 정의와 누적밀도 웨이블릿 히스토그램의 구조를 설계하고, 생성 절차를 알아본다. 4장에서는 제안된 기법을 이용한 선택을 추정 과정을 기술한다. 5장에서는 여러 실험을 통해 이 논문에서 제안된 기법과 기존 히스토그램의 성능평가를 수행한 후, 6장에서 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

이 장에서는 기존의 공간 선택을 추정에서 발생하는 여러 문제점을 해결하기 위해 제안된 공간 히스토그램들에 대해 알아보고, 이 논문에 기반이 되는 웨이블릿 변환(wavelet transformation)의 특성에 대해 기술한다.

2.1 공간 히스토그램

차원이 확장함에 따라 요구되는 저장 공간은 커지게 된다. 만약 각 차원을 D_1, D_2, \dots, D_n 이라 할 때, 데이터의 총 크기는 $\prod_{i=1}^n |D_n|$ 로 상당히 커지게 된다. 따라서 데이

터베이스 시스템에서는 필요한 요약 데이터 정보를 한정된 저장 공간에 저장하기 때문에 총 크기에 비해 매우 적은 정보를 가지고 선택율을 추정하게 된다. 그러므로, 이러한 문제점을 해결하기 위해 차원 변환 기법과 압축 기법이 제안되었다. 차원 변환 기법은 선이나 영역 같은 공간 객체를 포인트로 변환하여 데이터 크기를 줄이는 방법이다[5]. 그러나 이 기법은 공간 객체간의 위상 관계를 표현하는데 많은 어려움을 갖는다. 따라서, 이러한 문제의 대안으로 많은 양의 데이터를 압축하여 저장공간을 줄이는 방법으로 웨이블릿 기반 기법[6-8]이 제안되었다. 특히, [8]의 기법은 공간 데이터의 분포가 균일하게 분포되어 있다는 가정을 기반으로 웨이블릿 기법을 적용하여 요약정보를 생성하고 있다. 그러나, 실제 데이터 셋은 균일한 분포가 아닌 편중된 분포를 가질 수 있다. 따라서, 질의가 편중된 분포를 갖는 버킷에 교차할 경우, 선택을 추정에 있어서 매우 큰 오차가 발생한다. 이러한 문제를 해결하기 위해 공간 분포를 고려한 공간 영역을 분할하는 MinSkew 히스토그램[2]이 제안되었다. MinSkew 히스토그램은 공간 데이터가 일정하게 분포되었다는 가정에 대한 문제점을 해결하기 위해 공간을 분할 시 공간 밀집도의 차이가 큰 축으로 분할해 가는 계층적인 분할 방법이다. 이 기법은 공간 밀집도가 큰 영역에 많은 버킷을 할당하여 편중된 공간

분포에 대한 선택을 추정에 높은 정확도를 보인다. [9,10,12]의 압축히스토그램 기법은 MinSkew와 웨이블릿을 결합한 기법으로, 적은 공간의 사용으로 좋은 성능을 얻을 수 있지만, MinSkew 기법의 영역객체가 갖는 다중 계산문제를 해결하지 못하고 있다.

이러한 영역객체의 다중 계산문제는 객체 영역의 크기가 큰 객체에 대해 여러 그리드 셀에 중복계산 문제이다. 따라서, 큰 영역의 객체를 많이 포함하는 경우, 높은 오차를 갖는 선택율이 추정된다. 이러한 문제의 해결을 위한 히스토그램 기법들이 제안되었으며, 그 중 누적밀도 히스토그램과, 오일러 히스토그램이 높은 정확도를 갖는 대표적인 히스토그램 기법이다[3,4,11].

[3,11]의 누적밀도 히스토그램은 공간 객체를 최소경계사각형(MBR : Minimum Boundary Rectangle)의 각 좌상단점, 우상단점, 좌하단점, 우하단점에 대한 네 개의 누적 서브 히스토그램을 생성하여 중복계산 문제를 해결한다. 이 기법은 공간 질의에 대한 매우 빠른 처리와 높은 정확도를 갖는다. 그러나, 이 기법은 누적에 소요되는 비용과 네 개의 서브히스토그램을 각각 유지해야 하는 저장공간에 대한 제약사항을 갖는다.

[4]의 오일러 히스토그램은 그래프 이론의 수학적인 공식을 사용하여, 그리드 셀 사이의 교차점(vertice), 경계(edge), 면(face)과 교차하는 객체를 카운트하여 오일러 방정식을 통해 선택율을 추정한다. 여기서 도메인의 크기가 $N * N$ 이면 $(2N-1)^2$ 의 총 데이터 크기를 갖게 된다. 이것은 누적밀도 히스토그램과 비교하여 $4N-1$ 만 큼의 적은 저장공간을 사용하지만, 요구되는 저장공간의 크기를 만족하기에는 부족하다. 또한, 이 기법은 그리드 셀과 교차하는 모든 교차점, 경계, 면에 버킷을 할당하여 선택율을 추정함으로써, 점 객체에 대한 선택을 추정에 적용하기 어렵다.

이러한 기법들은 매우 정확한 선택율을 얻지만 작은 저장공간에서 이러한 히스토그램을 유지하기는 매우 어렵다. 그 중, 누적밀도 히스토그램은 유지되는 서브히스토그램의 데이터 분포가 매우 균일하게 분포하는 특징을 가진다. 웨이블릿 변환은 데이터 분포가 균일할 때, 뛰어난 압축 효과를 얻을 수 있다. 따라서 이 논문에서는 누적밀도 히스토그램의 저장공간에 대한 제약사항을 웨이블릿 변환 기법을 통하여 해결한다.

2.2 웨이블릿 변환

웨이블릿 변환 기법은 뛰어난 압축효과와 근사처리 능력으로 인해 주로 영상처리나 신호처리에 이용된다. 최근, 이 기법은 데이터베이스의 근사 질의(approximate query) 처리와 선택을 추정 기법에 이용되고 있다.

Haar 웨이블릿은 배율함수(scaling function) 계수와 웨이블릿(wavelet) 계수를 구하는데 평균함수(average

function)와 상세함수(detail function)를 사용한다[13]. 여기서, 상세함수에 의해 생성된 값을 웨이블릿 계수라 하며, 이 계수들을 가지고 웨이블릿 요약정보를 구성한다. 원본데이터의 복원과정은 두 상세계수에 대한 합과 차를 이용하여 웨이블릿 분해의 반대되는 과정을 통해 이루어진다.

웨이블릿 변환에서 생성된 웨이블릿 계수는 전체 데이터의 수와 같은 크기를 갖기 때문에 압축효과를 얻기 위해 웨이블릿 계수 축소 과정이 요구된다. 웨이블릿 변환 과정에서는 웨이블릿 변환의 대상이 되는 전체 데이터의 순서와 크기에 따라 계수들의 값이 '0'이거나 '0'에 가까운 값을 가지게 된다. 여기서, 값이 '0'인 계수들은 데이터 복원에 영향을 주지 않으므로 저장할 필요가 없다. 따라서, 값이 '0'인 계수가 많을수록 웨이블릿 압축효과가 높게 나타난다. 또한 '0'에 가까운 계수를 '0'으로 대체함으로써 추가적인 압축효과를 얻을 수 있다. 원본 데이터를 복원하기 위해서는 전체 데이터 복원에 영향을 최소화하는 계수 임계치(coefficient threshold)를 결정해야 한다. 데이터 복원에 대한 오차 계산법으로는 일반적으로 평균 제곱 에러 L^2 를 사용한다. L^2 를 최소화하기 위해서는 각 웨이블릿 계수에 대해 $\sqrt{2^{level}}$ 로 나누어진 정규화된 값이 큰 순서로 저장공간의 크기만큼 계수를 유지하고 나머지는 '0'으로 대체한다. 이러한 방법으로 웨이블릿 압축을 수행한다. 추정 오차를 최소화하여 압축시키는 방법으로는 결정적 임계(deterministic threshold)를 이용하는 방법과 상대 오차를 최소화하는 확률적 임계(probabilistic threshold)를 이용하는 방법이 있다[7]. 확률적 임계를 이용하는 방법은 신뢰성 있는 근사값을 얻을 수 있지만, 높은 계산 비용이 요구된다. 결정적 임계를 이용하는 방법은 L^2 오차를 최소화하는 방법으로 전체 오차를 최소화하는데 사용되며, 계산 비용이 저렴하다. 또한, 결정적 임계로 압축된 웨이블릿 요약정보는 범위 질의에 높은 정확도를 갖는다. 그러나 점 질의에 대해서는 큰 오차를 가지며, 추정된 값의 분포가 편중될 수 있다. 따라서, 이 논문에서는 범위 질의에 높은 정확도를 보이며, 계산 비용이 적은 결정적 임계 방법을 사용하여 웨이블릿 압축을 수행한다.

3. 누적밀도 웨이블릿 히스토그램

앞 장에서 언급한 내용과 같이, 이 논문의 목적은 최소의 저장공간의 사용으로 공간 객체의 특성을 충분히 유지하면서 높은 정확도의 선택율을 얻는 것이다. 기존의 연구로부터 우리는 누적밀도 히스토그램에 대한 각 서브히스토그램이 일정하게 증가하는 누적된 카운트 값을 유지하게 되며, 중복된 값이 여러 버킷에 중복되어

저장됨으로써, 유사한 공간 빈도를 갖는 이웃하는 버킷들이 많다는 사실을 발견하였다. 이러한 특징은 균일한 분포의 데이터에 대해 뛰어난 압축효과를 얻을 수 있는 웨이블릿 변환의 특징과 부합한다. 따라서, 이 논문에서는 누적밀도 히스토그램의 저장공간에 대한 제약사항을 압축기법인 웨이블릿 변환을 통해 해결함으로써, 이 논문의 목적을 만족하는 누적밀도 웨이블릿 히스토그램을 제안한다. 이 장에서는 제안된 공간 히스토그램의 생성 절차 및 구조와 선택을 추정 과정에 대해 기술한다.

3.1 표기

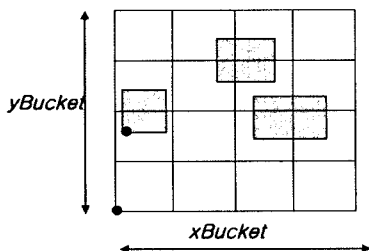
다음 표는 누적밀도 웨이블릿 히스토그램의 생성절차와 선택을 추정과정을 설명하기 위해 사용되는 표기를

표 1 웨이블릿 변환 위한 표기

표 기	의미
A_i, W_i	입력 데이터 배열과 웨이블릿 계수의 배열
B_i	그리드 셀에 대한 버킷
D_i	데이터 배열의 셀 i 에 데이터 값
O_i	셀 i 에 복원(근사) 데이터 배열
r_i	웨이블릿 계수 Coef.의 해상도 레벨
$Wav.coeff_i$	정규화 되지 않은 웨이블릿 계수
$Norm.coeff_i$	정규화된 웨이블릿 계수
$Retained\ coeff_i$	유지되는 웨이블릿 계수의 수

표 2 누적밀도 웨이블릿 히스토그램을 위한 표기

표 기	의미
Q	(qx_l, qy_l, qx_h, qy_h) 좌표를 갖는 질의 윈도우
BQ	질의 Q 와 교차하는 버킷
$xBucket$	x 축 버킷의 전체 크기
$yBucket$	y 축 버킷의 전체 크기
Llp	객체의 좌하단 점
Lrp	객체의 우하단 점
Hlp	객체의 좌상단 점
Hrp	객체의 우상단 점
$Hll(i, j)$	셀 (i, j) 까지 Llp 를 누적 카운트한 히스토그램
$Hlr(i, j)$	셀 (i, j) 까지 Lrp 를 누적 카운트한 히스토그램
$Hul(i, j)$	셀 (i, j) 까지 Hlp 를 누적 카운트한 히스토그램
$Hur(i, j)$	셀 (i, j) 까지 Hrp 를 누적 카운트한 히스토그램



다. 표 1은 웨이블릿 변환에서 사용되는 표기이며, 표 2는 누적밀도 웨이블릿 히스토그램에서 사용되는 표기를 나타낸다.

3.2 누적밀도 웨이블릿 히스토그램의 구성

우리는 누적밀도 히스토그램의 저장공간에 대한 제약사항을 줄이기 위해 웨이블릿 변환기법을 적용하였으며, 이 두 기법의 결합을 통해 선택율의 정확성과 저장공간의 제약사항을 동시에 해결할 수 있음을 발견하였다. 따라서, 이 논문에서 제안된 누적밀도 웨이블릿 히스토그램은 두 기법의 장점을 충분히 고려하여 설계한다.

3.2.1 구조 및 생성절차

누적밀도 웨이블릿 히스토그램은 다음과 같은 과정을 통하여 이루어진다.

1. 누적밀도 히스토그램 생성 : 전체 영역 $|D_x| * |D_y|$ 을 일정한 크기의 그리드 셀로 공간 분할한 후, 각 그리드 셀에 대한 버킷 B_i 의 크기를 결정한다. 각 버킷 B_i 의 크기가 결정되면, 객체의 각 코너점(Llp , Lrp , Hlp , Hrp)에 대한 위치를 결정한다. 그리고 전체 영역에 대한 각 객체의 각 코너 점을 누적해서 유지하는 4개의 서브히스토그램을 생성한다. 그림 1은 객체의 좌하단 점(Llp)을 누적 카운트한 Hll 히스토그램을 나타내고 있다. 누적밀도 히스토그램은 다음과 같은 구조를 갖는다.

$$CD\ Histogram = \langle bucket\ range, Hll, Hlr, Hul, Hur \rangle$$

$$bucket\ range = \langle P_L \in \{x_l, y_l\}, P_U \in \{x_h, y_h\} \rangle$$

$bucket\ range$: 각 버킷 범위를 나타내는 정보
 Hll, Hlr, Hul, Hur : 객체의 각 코너점에 대한 누적 카운트 값

$\{x_l, y_l\}, \{x_h, y_h\}$: 각 버킷의 좌하단 셀과 우상단 셀의 쌍

2. 웨이블릿 변환 : 이 단계에서는 생성된 각 네 코너점(Llp, Lrp, Hlp, Hrp)에 대한 누적 카운트한 2차원 버킷을 공간 탐색 기법을 사용하여 1차원 배열로 변환한다. 변환된 데이터를 입력 값으로 웨이블릿 변환을 수행한다. 이 논문에서는 개념적으로 가장 간단한

Hll

1	2	3	3
1	2	3	3
1	1	2	2
0	0	0	0

그림 1 좌하단 점(Llp)에 대한 서브히스토그램(Hll)

고, 빠른 계산 속도와 효율적인 저장공간을 갖는 1차원 Haar 웨이블릿을 사용하여 웨이블릿 요약정보 W_i 를 생성한다.

3. 웨이블릿 계수 축소(wavelet coefficient reduction) : 각 히스토그램에 대한 웨이블릿 요약정보 W_i 에 유지되는 웨이블릿 계수의 수를 한정된 저장공간의 크기만큼 축소하여 유지한다.

$B = \langle \text{Wavelet synopsis} \in (\text{coefficient}, \text{coefficient index}) \rangle$

여기서, 요구되는 버킷의 크기는 B이고, 각 버킷에 유지되는 웨이블릿 계수의 수를 W라 할 때, 각 구조의 인자들이 모두 같은 크기라면, 전체 저장공간의 크기 TM은 다음과 같이 정의된다.

$$TM = 4B(W)$$

3.2.2 누적밀도 히스토그램 생성

누적밀도 히스토그램은 다음과 같은 과정을 통하여 생성된다. 먼저, 전체 공간 영역은 일정한 크기의 그리드 셀로 분할하며, 그리드 셀과 동일한 수의 버킷을 할당한다. 각 버킷은 각 누적 서브 히스토그램(Hll, Hlr, Hul, Hur) 정보를 저장하며, Hll, Hlr, Hul, Hur에는 각 그리드 셀(i,j)에 대하여 다음과 같이 4개의 코너점에 대한 정보를 저장한다.

- 그리드 셀(0,j)에서 (i,j)까지 범위에 놓인 사각형 객체의 좌하단(lower_left corner) 점의 수를 $LL(i, j)$ 이라 할 때, $Hll(i, j)$ 는 (0,j)에서 (i,j)까지의 범위에서 좌하단 점을 갖는 객체의 수를 누적해서 저장하게 되며, (1)과 같이 계산된다.

$$Hll(i, j) = \sum_{x=0}^j LL(i, x) \quad (1)$$

- 그리드 셀(0,j)에서 (i,j)까지 범위에 놓인 사각형 객체의 우하단(lower_right corner) 점의 수를 $LR(i,j)$ 이라 할 때, $Hlr(i, j)$ 는 (0,j)에서 (i,j)까지의 범위에서 좌하단 점을 갖는 객체의 수를 누적해서 저장하게 되며, (2)과 같이 계산된다.

$$Hlr(i, j) = \sum_{x=0}^j LR(i, x) \quad (2)$$

- 그리드 셀(0,j)에서 (i,j)까지 범위에 놓인 사각형 객체의 좌상단(upper_left corner) 점의 수를 $UL(i,j)$ 이라 할 때, $Hul(i,j)$ 는 (0,j)에서 (i,j)까지의 범위에서 좌하단 점을 갖는 객체의 수를 누적해서 저장하게 되며, (3)과 같이 계산된다.

$$Hul(i, j) = \sum_{x=0}^j UL(i, x) \quad (3)$$

- 그리드 셀(0,j)에서 (i,j)까지 범위에 놓인 사각형 객체의 우상단(upper_right corner) 점의 수를 $UR(i,j)$ 이라 할 때, $Hur(i,j)$ 는 (0,j)에서 (i,j)까지의 범위에서

좌하단 점을 갖는 객체의 수를 누적해서 저장하게 되며, (4)와 같이 계산된다.

$$Hur(i, j) = \sum_{x=0}^j UR(i, x) \quad (4)$$

3.2.3 웨이블릿 요약정보

모든 누적밀도 히스토그램 구성이 끝나면, 생성된 각 서브 히스토그램(Hll, Hlr, Hul, Hur)을 웨이블릿 변환을 통해 압축하여 유지한다. 먼저, 각 서브 히스토그램에 대해 그리드 셀에 대응하는 2차원 버킷을 1차원으로 변환하는 과정이 필요하다. 이 과정은 공간 탐색 기법을 통해 이루어진다. 공간 탐색 기법에는 Hilbert Order, Z-Order 등 여러 기법들이 있으며, 그 중 웨이블릿 변환을 적용할 때, 가장 효율적인 압축효과를 고려하여 선택한다.

웨이블릿은 인접한 영역의 값이 유사할 경우, '0'에 가까운 계수가 많이 생성되며, '0'에 가까운 계수가 많을수록 압축효과가 큰 것을 의미한다. 따라서, 웨이블릿 변환은 공간 분포가 균일할 때, 최대의 압축효과를 얻을 수 있다. 누적밀도 히스토그램은 누적 카운트 값이 일정하게 증가하여 구성되며, 같은 값을 여러 버킷에 유지함으로써, 많은 저장공간이 요구된다. 그러므로, 이 논문에서는 유지되는 네 서브히스토그램의 인접성을 고려하여 인접한 버킷 탐색에 유리한 Z-Mirror Order 기법을 사용한다.

각 유지되는 서브히스토그램은 1차원 Haar 웨이블릿 변환을 적용하기 위해 2차원 버킷을 1차원 배열 형태로 변환한다. 여기서 변환된 1차원 배열을 이용하여 웨이블릿 요약정보를 생성한다. 알고리즘 1은 각 서브 히스토그램(Hll, Hlr, Hul, Hur)에 대해, Z-mirror curve에 의해 1차원 변환된 배열의 데이터 집합 A_i 를 웨이블릿 변환하는 과정을 보여주는 알고리즘이다.

알고리즘 1 웨이블릿 변환(Haar Wavelet Transformation)

Input : 각 서브 히스토그램(Hll, Hlr, Hul, Hur)에 대해, Z-mirror curve에 의해 1차원 변환된 배열의 데이터 집합 (A_i), A_i 의 인덱스의 크기(IndexSize), 해상도 레벨(r)

Begin

$r = 0$; // 해상도 레벨 초기화

While $r < \log_2 \text{IndexSize}$

For IndexSize in A_i Do

// A_i 의 평균 값(Average value)을 계산한다.

$\text{Avg}_i = (A_{2i} + A_{2i+1}) / 2$;

// A_i 의 상세 계수 (Detail coefficient)를 계산한다.

$\text{Det}_i = (A_{2i} - A_{2i+1}) / 2$;

End For

$r = r + 1$; //해상도 레벨은 증가한다.

// 계산된 평균 값(Average value)과 상세 계수(Detail coefficient)를 웨이블릿 시놉시스(Wavelet synopsis)에 유지한다.

```

For IndexSize Do
    Wavg = Avgi
    Wdet = Deti
    Ai = Wavg
EndFor
EndWhile
End
    
```

그림 2(a)는 그림 1에서의 객체의 좌하단 점(Lip)에 대해 생성된 2차원 버킷의 Hll 히스토그램을 Z-Mirror Order를 사용하여 1차원 배열로 변환하는 과정을 나타낸다. 즉, Hll 히스토그램 버킷 카운트 값은 그림 2(a)와 같이 1, 2, 1, 2, 1, 1, 0, 0 등의 순서로 1차원 배열에 저장된다. 그림 2(b)는 그림 2(a)에 의해 변환된 1차원 배열의 데이터 집합을 입력 값으로 사용하여, 웨이블릿 변환을 수행하는 과정을 통해 생성된 오차 트리를 보여준다. 즉, 원본데이터 O₁, O₂의 평균값은 (1+2)/2 = 1.5, 상세계수는 (1-2)/2 = -0.5로 계산되며, O₃, O₄의 평균값은 (1+2)/2 = 1.5, 상세계수는 (1-2)/2 = -0.5로 계산된다. 이와 같이 계산된 level 4에 대한 평균값을 가지고, 다음 레벨(level 3)의 평균값과 상세계수를 반복적으로 계산함으로써, 그림 2(b)와 같은 오차트리가 생성된다. 그림 2(b)에서 웨이블릿 변환을 통해 생성된 웨이블릿 계수는 전체 데이터의 수 O_i와 같은 크기를 갖기 때문에 압축효과를 얻기 위해 웨이블릿 계수 축소 과정이 요구된다. 여기서, 값이 '0'인 계수들은 데이터 복원에 영향을 주지 않으므로 저장하지 않으며, '0'에 가까운

계수를 '0'으로 대치함으로써 추가적인 압축효과를 얻을 수 있다.

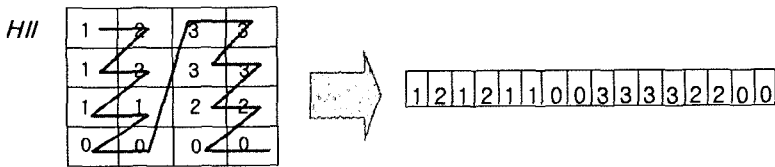
3.2.4 웨이블릿 계수 축소

각 웨이블릿 요약정보(Wav.coeff)에 유지되는 계수는 한정된 저장공간을 만족할 때까지 축소한다. 예를 들어, 원본데이터가 필요로 하는 저장공간이 16일 때, 한정된 저장공간을 8이라 하자. 그림 3에서 생성된 웨이블릿 요약정보의 원본데이터는 16만큼의 저장공간이 요구된다. 이 요약정보는 한정된 저장공간 8을 만족시키기 위해 결정적 임계(deterministic threshold)를 사용한다. 각 계수는 $\sqrt{2^{Level}}$ 로 나누어진 정규화된 값이 큰 순서로 저장공간의 크기만큼 계수를 유지하고 나머지는 '0'으로 대치한다.

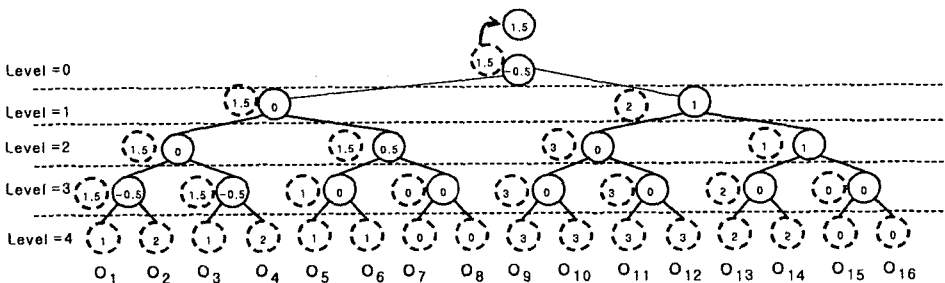
그림 3(a)는 Hll 히스토그램에 대한 웨이블릿 시놉시스(wavelet synopsis)를 나타내며, 여기서 웨이블릿 계수들에 대한 정규화된 계수(Norm.coeff)를 확인할 수 있다. 따라서, 한정된 저장공간 8에 대하여 유지되는 웨이블릿 계수는 W_i={1.5, -0.5, 0.5, 1, 0.5, 1, -0.5, -0.5}이며, 나머지 계수는 모두 '0'으로 대치됨으로써, 압축효과를 얻는다. 그림 3(b)는 한정된 저장공간이 6일 경우, 유지되는 웨이블릿 계수와 오차 트리를 보여준다.

4. 선택을 추정

이 장에서는 앞 장에서 기술한 누적밀도 웨이블릿 히스토그램의 요약정보를 이용하여 선택을 추정하는 과정을 설명한다.



(a) Hll 히스토그램의 1차원 변환 과정

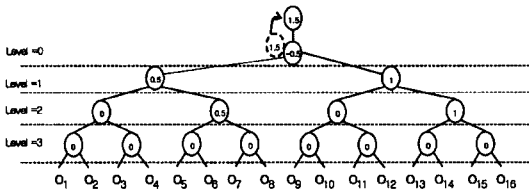


(b) 1차원 배열에 대한 오차 트리

그림 2 누적밀도 웨이블릿 히스토그램의 웨이블릿 변환

Index l	0	1	2	3	4	5	6	7
	8	9	10	11	12	13	14	15
Wav. coeff	1.5	-0.5	0.5	1	0	0.5	0	1
	-0.5	-0.5	0	0	0	0	0	0
Level coeff	0	0	1	1	2	2	2	2
	3	3	3	3	3	3	3	3
Norm coeff	1.5	0.5	0.5	1	0	0.5	0	1
	$\frac{0.5}{\sqrt{2}}$	$\frac{0.5}{\sqrt{2}}$	0	0	0	0	0	0

(a) 웨이블릿 시뮬시스



(b) 한정된 저장공간에 대한 오차 트리 그림 3 웨이블릿 계수의 정규화 및 축소

임의의 질의 $Q(qxl, qxh, qyl, qyh)$ 가 주어지면, 질의 윈도우 Q 에 대한 특정 버킷의 인덱스를 공간 탐색 기법에 의해 변환된 각 서브 히스토그램(Hll, Hlr, Hul, Hur)의 1차원 배열에서 찾는다. 찾은 각 서브히스토그램에 대한 특정 버킷의 인덱스는 웨이블릿 복원과정을 통해 인덱스에 해당하는 원본데이터를 복원한다. 여기서 복원된 원본데이터를 이용하여 선택을 구한다. 따라서, 누적밀도 웨이블릿 히스토그램의 경우, 기존 누적밀도 히스토그램에 비해 추정시 웨이블릿 복원을 위해 $\log N+1$ 시간이 더 걸리게 된다. 그러나, 약 50%정도의 저장공간 효율성을 지원할 수 있는 장점이 있다. 각 서브 히스토그램의 특정 버킷을 복원하는 과정은 알고리즘 2에서 자세히 기술된다.

알고리즘 2 원본데이터에 대한 웨이블릿 복원 과정

```

Input : 웨이블릿 계수(Deti), 오차 트리(Error Tree)의 인덱스(Coeffi), 해상도 레벨(r), 원본 데이터의 인덱스(Oi)
Output : 질의에 대한 각 서브히스토그램의 원본데이터(OrDatai)

각 서브히스토그램(Hll, Hlr, Hul, Hur)에 다음과 같은 과정을 수행한다.
Begin
  Coeffp = 0; //오차 트리의 현재 노드
  Coeffc = 0; //오차 트리의 자식 노드
  Pathi = 0; //오차 트리에서의 복원 경로를 저장하기 위한 배열
  While Each histogram
    For Coeffi Do
    
```

```

// 만약 Coeffp와 그 자식 레벨의 Coeffc가 Coeff2p = Coeffc 이면 즉, 복원하는 데이터가 노드의 왼쪽이면 양수(+)값을 가지고 웨이블릿 계수를 저장한다.
  If Coeff2p = Coeffc ∈ Oi then
    Pathi = Deti
// 만약 Coeffp와 그 자식 레벨의 Coeffc가 Coeff2p+1 = Coeffc 이면 즉, 복원하는 데이터가 노드의 오른쪽이면 음수(-)값을 가지고 웨이블릿 계수를 저장한다.
  Else If Coeff2p+1 = Coeffc ∈ Oi then
    Pathj = -Deti
  End If
End For
OrDatai = Pathi + Pathj
End While
End
    
```

다음 예를 통하여, 이 논문에서 제안된 기법의 선택을 추정과정을 설명한다. 예를 들어, 그림 4는 임의의 질의 Q 에 대한 각 서브히스토그램을 나타낸다. 여기서 각 서브히스토그램의 Hll [qxh,qyh], Hlr [qxl-1,qyh], Hul [qxh,Wyl-1], Hur [qxl-1,qyl-1]에 해당하는 각 버킷의 인덱스를 공간 탐색 기법에 의해 변환된 1차원 배열에서 찾는다.

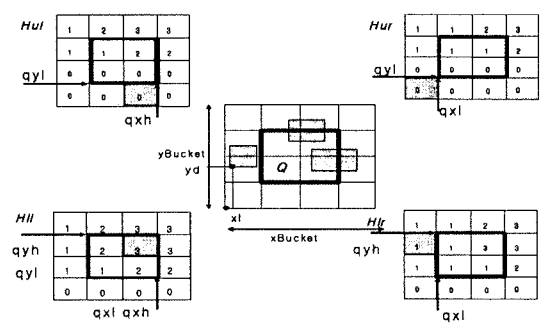


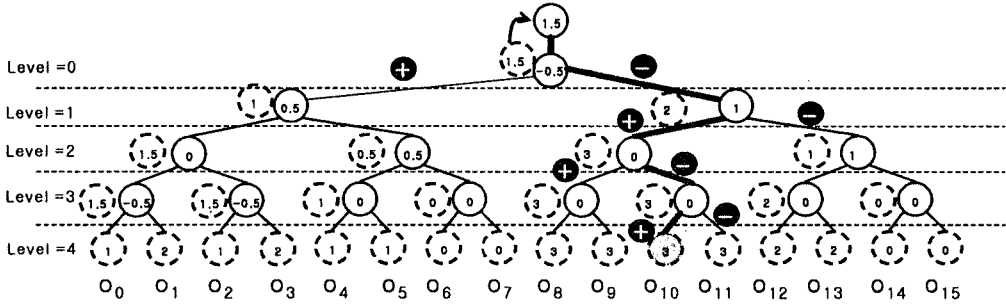
그림 4 임의의 질의 Q 에 대한 Hll 히스토그램

그림 5(a)는 Hll 히스토그램에 대한 1차원 배열을 나타내며, 질의 윈도우 Q 에 대한 Hll [qxh,qyh]에 해당하는 인덱스 O_{10} 을 여기서 확인할 수 있다. 그림 5(b)는 O_{10} 의 원본데이터 복원 과정을 보여주고 있다. 복원은 root부터 시작해서 복원하기 위한 데이터가 노드의 왼쪽이면 (+), 오른쪽이면 (-) 부호를 갖는다. 즉, 복원하기 위한 원본 데이터 O_{10} 까지의 경로 안에 존재하는 모든 노드에 대해 (5)와 같이 계산함으로써, Path(O_{10})를 구한다.

예를 들어, 그림 5(b)의 O_{10} 의 복원 과정을 살펴보면, root부터 시작해서 O_{10} 의 위치가 현재 노드의 왼쪽이면

Count value	1	2	1	2	1	1	0	0	3	3	3	3	2	2	0	0
OrIndex O_i	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}

(a) Z-Mirror Order를 통한 1차원 변환



(b) 원본데이터에 대한 오차 트리

그림 5 선택을 추정할 위한 웨이블릿 오차 트리의 복원 예

계수는 양수(+) 값의 부호를 갖고, 오른쪽이면 계수는 음수(-) 부호를 갖는다. 즉, O_{10} 의 $Path(O_{10}) = 1.5 - (-0.5) + 1 - 0 + 0 = 3$ 으로 원본 데이터는 복원된다.

$$o_i = \sum_{coeff_j \in path(o_i)} dir \cdot coeff_j, \text{ if } dir \text{ is } o_i \\ \in \text{Leftleaves}(coeff_j) \text{ or } j=0 \text{ then } +1 \text{ otherwise } -1 \quad (5)$$

각 서브히스토그램(Hll, Hlr, Hul, Hur)은 위와 같은 과정을 통하여 특정 위치의 버킷 카운트 값을 복원하여 $Hll [qxh, qyh] = 3, Hlr [qxl-1, qyh] = 1, Hul [qxh, Wyl-1] = 0, Hur [qxl-1, qyl-1] = 0$ 값을 얻는다. 이렇게 복원된 값을 이용하여 질의 윈도우 Q에 대한 선택을 다음과 같은 공식을 통해 구한다.

$$\text{선택율} = Hll [O_{10}] - Hlr [O_2] - Hul [O_{14}] + Hur [O_6] \\ = 3 - 1 - 0 + 0 = 2$$

5. 실험 및 평가

이 장에서는 누적밀도 웨이블릿 히스토그램의 효율성을 보이기 위해 질의 크기 및 저장공간의 크기에 따른 평균 상대 오차율과 그리드 레벨의 증가에 따른 평균 상대 오차율을 측정하여 선택율의 정확성을 실험평가하고, 그 결과를 제시한다.

5.1 실험 환경 및 평가 기준

이 논문에서 제안된 기법의 성능 평가를 위해 512MB의 주 메모리와 80GB 하드디스크, 윈도우 2000 서버하의 Intel Pentium III 1.0GHz PC에서 실험하였다. 실험 데이터 셋으로는 2,249,727개의 TIGER/LINE 캘리포니아 데이터셋(Dataset 1), Sequoia 2000 벤치마크를 위한

22,288개의 실제 폴리곤 데이터셋(Dataset 2), 10,484개의 서울시 중구의 빌딩 데이터셋(Dataset 3)을 대상으로 실험하였다. 또한, 질의 변화에 따른 성능을 평가하기 위해 5%, 10%, 15%, 20%의 질의 크기의 변화를 주었으며, 유지되는 웨이블릿 계수의 수를 다르게 하여, 즉, 저장공간 크기를 다르게 설정하여, 저장공간 크기에 따른 누적밀도 웨이블릿 히스토그램의 성능을 평가하였다.

이 실험데이터 집합의 공간 도메인은 (Dataset 1) = [(114.12700, 124.50000), (32.400000, 42.10920)], (Dataset 2) = [(1150000, 2030000), (10869, 1320000)], (Dataset 3) = [(196500, 202300), (449500, 452100)]이며, 임의의 10개의 질의에 대한 추정 결과들에 대해 평균을 구하여 오차율을 측정하였다.

이 실험에서 우리는 저장공간에 따른 평균 상대 오차율을 평가하기 위해 한정된 저장공간의 제약사항을 두어 실험하였다. 즉, 기본 버킷수가 100인 경우, CD 히스토그램에서 요구되는 단위 저장공간은 800이며, CDW 히스토그램에서는 $CDW_1 = 400, CDW_2 = 266, CDW_3 = 200$ 의 각 저장공간의 크기를 설정하여, CD 히스토그램의 저장공간의 각 50%, 33%, 25%의 사용으로 CD 히스토그램과의 평균 상대 오차율을 비교평가하였다.

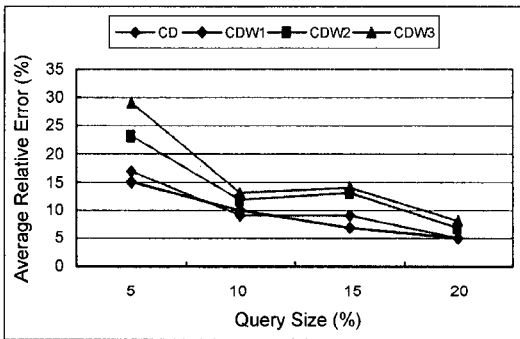
선택율 추정 결과의 평가는 오차를 측정하기 위해 평균 상대 오차를 사용하였다. 평균 상대 오차 Er 은 질의 집합 Q에 속하는 임의의 질의 q에 대해 실제 결과 크기가 Nq 이고, 추정 결과 크기가 $N'q$ 일 때, (6)으로 정의된다.

$$Er = \sum_{q \in Q} \frac{|Nq - Nq'|}{Nq} \times 100 \quad (6)$$

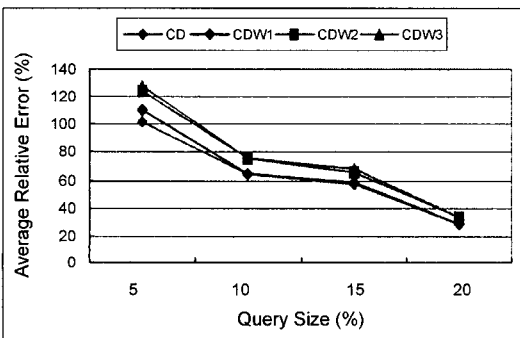
5.2 실험 결과 및 평가

5.2.1 질의 및 저장공간에 따른 평가

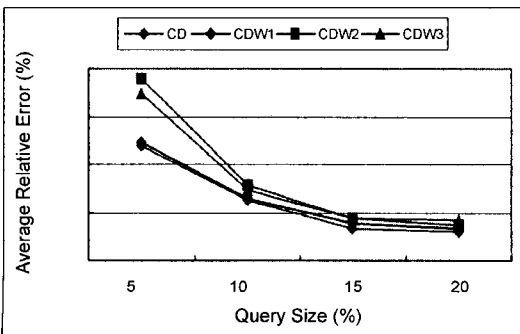
그림 6(a)는 CDW 히스토그램 CDW1, CDW2, CDW3 와 CD 히스토그램을 각 5%, 10%, 15%, 20% 질의를 통해 얻은 평균 상대 오차율의 평균 오차율에 대한 실험 결과이다. 이 그림에서 질의 크기에 대한 CDW1, CDW2, CDW3의 평균 상대 오차율의 변화를 살펴보면,



(a) TIGER/LINE Dataset



(b) Sequoia 2000 벤치마크 Dataset



(c) 서울시 중구 폴리곤 Dataset

그림 6 질의 크기에 따른 평균 상대적 오차율

전체적으로 질의의 크기가 증가할수록 선택율의 정확도는 증가한다. 그리고 CD 히스토그램 저장공간 크기의 33%와 25%를 사용하는 CDW2, CDW3는 50%를 사용하는 CDW1과 비교하여 높은 오차의 선택율을 보이는 것으로 나타났다. 크기가 작은 질의는 상대적으로 큰 버킷 영역에 포함되기 쉽기 때문에 높은 오차를 발생시킨다. 그러나 유지되는 웨이블릿 계수가 큰 CDW1은 버킷내의 공간 분포 정보를 웨이블릿 요약정보로 함축되어 저장하고 있기 때문에 보다 낮은 오차를 갖는다. 또한, 크기가 큰 질의는 하나의 버킷을 완전히 포함하거나, 대부분의 버킷을 포함하기 때문에 좀 더 높은 정확도의 선택율을 얻을 수 있다.

그림 6(b)와 그림 6(c)는 각 데이터셋(Dataset 2, 3)에 대한 평균 상대 오차율을 보이고 있다. 그림 6(b)에서 보여주듯이 CDW1을 제외한 CDW2, CDW3는 기존 CD보다 약 20% 높은 오차율을 보였으며, 그림 6(c)에서는 질의 크기의 증가에 따라 모든 CDW 히스토그램이 유사한 평균 상대적 오차율을 보이고 있다.

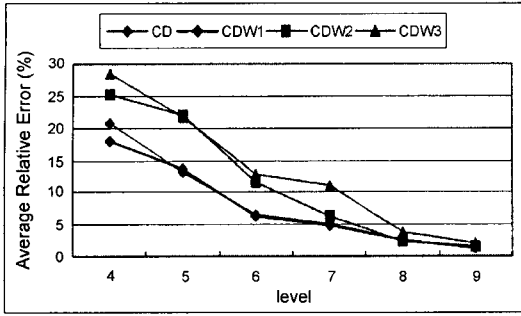
여기서, 작은 저장공간의 사용은 웨이블릿 계수를 저장할 공간을 절약하고, 복원에 사용되는 계수가 적어져 복원 시 계산량을 감소 시킨다. 그러나, 반대로 웨이블릿 압축으로 무시되는 계수는 오차로 작용하여 복원 시 오차가 발생한다. 따라서, 정확도는 최대로 유지하면서 높은 압축 효과를 얻는 방향으로 웨이블릿 압축이 수행되어야 한다. 이 실험에서는 기존 CD 히스토그램 저장공간의 50%만을 사용한 CDW1이 최소의 저장공간의 사용으로 최대의 정확성을 보이는 것으로 나타났다.

5.2.2 그리드 레벨에 따른 평가

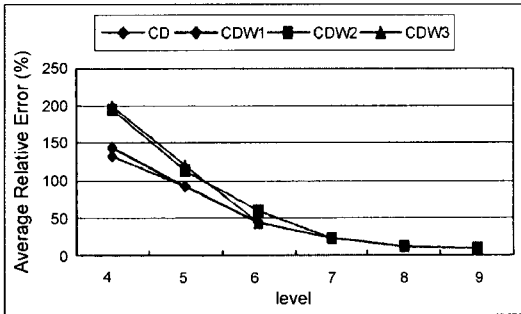
이 실험에서는 각 그리드 레벨에 따라 저장공간의 크기를 기존 CD 히스토그램에 CDW1 = 50%, CDW2 = 33%, CDW3 = 25%의 저장공간을 사용하였으며, 그리드의 레벨을 4~9(16 x 16~512 x 512)의 변화를 주어, 그리드 레벨에 따른 평균 상대적 오차율을 비교평가 하였다. 그림 7(a)는 Dataset 1에 대한 CD 히스토그램과 CDW 히스토그램의 그리드 레벨에 따른 평균 상대적 오차율을 보이고 있다.

이 그림에서 보여주듯이 CD 히스토그램과 CDW1 히스토그램은 유사한 오차율을 보이고, CDW2, CDW3는 CD와 CDW1에 비해 높은 오차율을 보여주고 있다. 그러나, 그리드 레벨이 9인 경우에는 모든 히스토그램이 비슷한 성능을 보인다. 이러한 이유는 그리드 레벨이 상세화 될수록 질의 원도우가 포함하는 버킷의 수가 보다 많아지기 때문이다.

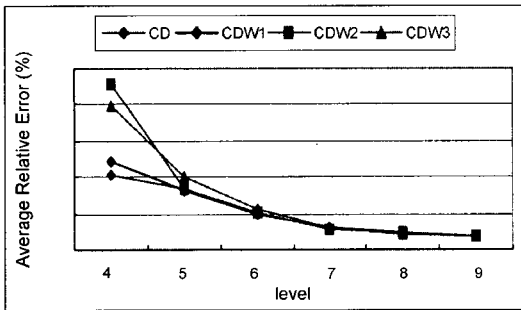
그림 7(b)와 그림 7(c)는 각 데이터 셋에 대한 그리드 레벨에 따른 평균 상대적 오차율을 보이고 있다. 그림 7(b)에서는 CDW1~CDW3 히스토그램 모두 유사한 평



(a) TIGER/LINE Dataset



(b) Sequoia 2000 벤치마크



(c) 서울시 중구 폴리곤 Dataset

그림 7 그리드 레벨에 따른 평균 상대적 오차율

균 상대적 오차 곡선을 보여주고 있으며, 이러한 실험 결과는 공간 데이터 분포에 따라 기존 CD 히스토그램이 갖는 저장공간에 최대 50%의 사용으로 타당한 선택을 얻을 수 있음을 보여주고 있다. 이 실험에서 각 히스토그램은 그리드 레벨이 증가함에 따라 선택율의 정확도가 증가하는 경향을 보이며, 그리드 레벨이 9일 경우에 5%이하의 비교적 정확한 평균 상대적 오차율을 보여주고 있다.

이 논문에서는 공간 분포를 적은 저장공간에서 유지함과 동시에 정확성 높은 선택율을 얻는데 그 목적이 있다. 따라서, 실험결과 최소의 저장공간의 사용으로 최

대의 정확성을 보이는 CDW1일 때, 최적의 선택율을 보이는 것으로 평가 되었다. 또한, 매우 제약적인 저장공간을 갖는 경우, 기존 기법의 최대 33%의 저장공간에 사용으로 비교적 정확한 선택율을 얻을 수 있음을 보였다.

6. 결론

대용량의 공간 데이터베이스 시스템에서는 효율적인 질의 처리를 위하여 요약정보를 구성하고, 이를 기반으로 질의 최적화를 위한 선택율을 추정하고 있다. 기존 공간 선택을 추정과 관련된 기존의 연구들은 대부분 요약정보를 위한 충분한 저장공간이 확보되었다는 가정을 기반으로 수행된다. 그러나 실제 데이터베이스 시스템에서의 요약정보는 제한된 저장공간을 사용하게 되며, 이러한 제한된 저장공간에서 많은 양의 공간 데이터에 대한 요약 정보를 생성할 경우, 공간 데이터의 특성을 반영하기 어렵다. 따라서, 선택을 추정시 많은 오차가 발생할 수 있다. 그러므로 저장공간의 제약사항을 감소시키기 위한 새로운 접근 방법이 필요하다. 따라서, 이 논문에서는 기존 공간 히스토그램의 저장공간에 대한 문제점을 해결하기 위하여 누적밀도 웨이블릿 히스토그램을 제안하였다. 제안된 누적밀도 웨이블릿 히스토그램은 웨이블릿 기법과 누적밀도 히스토그램의 장점을 채택하여, 보다 적은 저장공간에서 요약정보를 유지하고, 정확한 선택율을 얻을 수 있다. 실험을 통하여, 제안된 누적밀도 웨이블릿 히스토그램이 기존의 누적밀도 히스토그램보다 50%정도 적은 저장공간을 사용하면서도 좋은 선택율을 얻을 수 있음을 보였다.

향후 연구에서는 객체의 변화에 따른 동적 대응이 가능하도록 히스토그램 갱신에 관한 연구를 수행할 것이다.

참고 문헌

- [1] Poosala et al., "Improved Histograms for Selectivity Estimation of Range Predicates," In Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.294-305, 1996.
- [2] Swarup Acharya, Viswanath Poosala, Sridhar Ramaswamy, "Selectivity estimation in spatial databases," In Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.13-24, 1999.
- [3] Jin, N. An, A. Sivasubramaniam, "Analyzing Range Queries on Spatial Data," In Proceedings of the IEEE International Conference on Data Engineering (ICDE), pp.525-534, 2000.
- [4] C. Sun, D. Agrawal, A. El Abbadi, "Selectivity for spatial joins with geometric selections," Proc. of EDBT, pp.609-626, 2002.
- [5] 김홍연, 배혜영, "다차원 히스토그램을 이용한 공간 위상 술어의 선택도 추정 기법", 한국 정보처리학회 논

문지 제6권 제4호, pp.841-850, 1999.4.

- [6] Vitter, Wang, "Approximate Computation of Multidimensional Aggregates of Sparse Data using Wavelets," In Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.193-204, 1999.
- [7] Yossi Matias, Jeffrey Scott Vitter, Min Wang, "Dynamic Maintenance of Wavelet-Based Histograms," The VLDB Journal, pp.101-110, 2000.
- [8] Min Wang, Jeffrey Scott Vitter, Lipyeow Lim, Sriram Padmanabhan, "Wavelet-based cost Estimation for Spatial Queries," In Proc. Int. Symp. on Spatial and Temporal Databases, pp.175-196, 2001.
- [9] 지정희, 이진열, 김상호, 류근호, "압축된 공간 히스토그램을 이용한 선택을 추정 기법", 한국정보처리학회 논문지 D 제11권 제2호 pp.281-292, 2004.
- [10] Jeong Hee Chi, Jin Yul Lee, Keun Ho Ryu, "Selectivity Estimation for Spatial Databases," Asian Conference on Remote Sensing & International Symposium on Remote Sensing (ISRS), November 2003.
- [11] 지정희, 김상호, 류근호, "일반화된 누적밀도 히스토그램을 이용한 공간 선택을 추정", 한국정보처리학회 논문지 D 제11권 제4호 pp.963-990, 2004.
- [12] Jeong Hee Chi, Sang Ho Kim, Keun Ho Ryu, "Spatial Selectivity Estimation using Compressed Histogram Information," LNCS 3399, 489-494, 2005.3.
- [13] Stollnitz, E., DeRose, T., and Salesin, D., "Wavelet for Computer Graphics Theory and Applications," Morgan Kaufmann, 1996.



류 근 호

1976년 숭실대학교 전자계산학과 졸업
 1980년 연세대학교 공학대학원 전자계산학 석사. 1988년 연세대학교 대학원 전자계산학 박사. 1976년~1986년 육군군수 지원사전산실(ROTC 장교), 한국전자통신연구소(연구원), 한국방송통신대, 전산학과(조교수) 근무. 1989년~1991년 Univ. of Arizona 연구원(TempIS Project). 1986년~현재 충북대학교 전기전자 및 컴퓨터공학부 교수. 관심분야는 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 객체 및 지식베이스 시스템, 지식기반 정보검색시스템, 데이터 마이닝, 데이터베이스 보안 및 Bio-Informatics



지 정 희

1999년 충주대학교 전자계산학과 졸업
 2001년 충주대학교 대학원 전자계산학 석사. 2003년 충북대학교 대학원 전자계산학 박사수료. 관심분야는 시공간 데이터베이스, Temporal GIS, 스트림 데이터베이스



정 재 혁

2003년 서원대학교 전자계산학과. 2005년 충북대학교 대학원 전자계산학 석사
 2005년~현재 한국에너지기술연구원 근무. 관심분야는 시공간 데이터베이스, 절의 최적화, 신재생 에너지 자원 GIS 시스템 및 정보화