

# Multi-resolution Algorithms for Building Spatial Histograms

Qing Liu

Yidong Yuan

Xuemin Lin

School of Computer Science and Engineering  
University of New South Wales  
Sydney, 2052 NSW, Australia  
{qingl,yidong,lxue}@cse.unsw.edu.au

## Abstract

Selectivity estimation of queries not only provides useful information to the query processing optimization but also may give users a preview of processing results. In this paper, we investigate the problem of selectivity estimation in the context of a spatial dataset. Specifically, we focus on the calculation of four relations, *contains*, *contained*, *overlap* and *disjoint*, between data objects and a query rectangle using *Euler*-histograms. We first provide a multi-resolution algorithm which can lead to the exact solutions but at the cost of storage space. To conform to a given storage space, we also provide an approximate algorithm based on a hybrid multi-resolution paradigm. Our experiments suggest that our algorithms greatly out-perform the existing techniques.

*Keywords:* Approximation query processing, Spatial databases, Histograms, Selectivity estimation.

## 1 Introduction

Research in spatial database management systems has recently emerged as central to numerous important applications. These include geographic information systems, computer-aided design, robotics, image processing and very large scale integration. Selectivity estimation is one of the most important aspects to the success of a development of query processing optimizer in spatial database management systems. Moreover, techniques developed in selectivity estimation may be used for providing approximate query processing results (Garofalakis, Gehrke & Rastogi 2002) and giving user a preview (Garofalakis et al. 2002) of results before issuing more complex queries. In this paper, we will investigate the problem of selectivity estimation in very large spatial datasets. Specifically, we will investigate the problem of an effective estimation of the break-down information about the number of objects from a large spatial dataset, which have "contains", or "contained", or "overlap", or "disjoint" relation between the objects and a given query window.

There are many search and index techniques (Zhou, Truffet & Han 1999) (Xiao, Zhang & Jia 2001). Sampling (Lipton, Naughton & Schneider 1990) was the most popular technique to estimate the spatial selectivity. To overcome a possible skew distribution, histogram techniques (Acharya, Poosala & Ranmaswamy 1999, Abounaga & Naughton 2000, Jin, An & Sivasubramaniam 2000, Beigel & Tanin 1998) have been recently developed. In this paper, we will investigate several new spatial histogram techniques.

Copyright ©2003, Australian Computer Society, Inc. This paper appeared at Fourteenth Australasian Database Conference (ADC2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol. 17. Xiaofang Zhou and Klaus-Dieter Schewe, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

There are a number of recent histogram techniques for spatial selectivity estimation. To estimate the number of objects which *intersect* a given query rectangle, the authors in (Beigel & Tanin 1998) proposed to build a spatial histogram based on the *Euler* theorem (Harary 1969). It has been shown that the techniques developed in (Beigel & Tanin 1998) can guarantee the exact solutions if the given query rectangle is a rectangle aligning with the histogram "grid". The paper (Jin et al. 2000) provides another technique *Cumulative Density Algorithm* to target the same problem as in (Beigel & Tanin 1998). The Min-skew algorithm (Acharya et al. 1999) and the SQ-histogram technique (Abounaga & Naughton 2000) investigated the problem of effectively partitioning the data space to accommodate an arbitrary query rectangle. The authors in (Sun, Agrawal & Abbadi 2002) have gone one step further to identify finer spatial relations; that is, the *intersect* relation is decomposed into three relations, *overlap*, *contains*, and *contained* (to be precisely defined in the next section). Based on the Euler-histogram techniques in (Beigel & Tanin 1998), (Sun et al. 2002) presents 3 approximate algorithms to calculate these 3 finer relations together with the disjoint relation.

In this paper, we first present an exact algorithm based on a multi-resolution paradigm to identify the 4 relations: *overlap*, *contains*, *contained*, and *disjoint*. This is the first contribution of the paper. The second contribution of the paper is to provide a new approximate algorithm, based on a combination of spatial histogram techniques and statistic techniques, to conform to a given storage space. Both of our algorithms can run in a constant time. Our experiments showed that our approximate algorithm can greatly improve the accuracy compared with the previous techniques.

The rest of the paper is organized as follows. In Section 2, we provide a background overview together with the necessary notation. Section 3 presents our new techniques. Section 4 presents the experiment results. This is followed by conclusion and remarks.

## 2 Preliminary

A spatial object is usually encompassed by the minimal *isothetic* bounding rectangle (MBR) to approximate its spatial extent. The binary topological relation between two objects,  $D$  and  $Q$ , is based upon the comparison (Egenhofer & Herring 1994) of  $D$ 's *interior* ( $D_i$ ), *boundary* ( $D_b$ ), *exterior* ( $D_e$ ) (shown in Figure 1(a)) with  $Q$ 's interior ( $Q_i$ ), boundary ( $Q_b$ ), and exterior ( $Q_e$ ); it can be classified into 8 different relations. The four relations, as depicted in Figures 1(b) - (e), are regarded (Sun et al. 2002) as the most important ones.

In this paper, we are interested in a set of MBRs only. Consequently, a set  $S$  of objects always means a set of isothetic rectangles. The *Euler* theory was

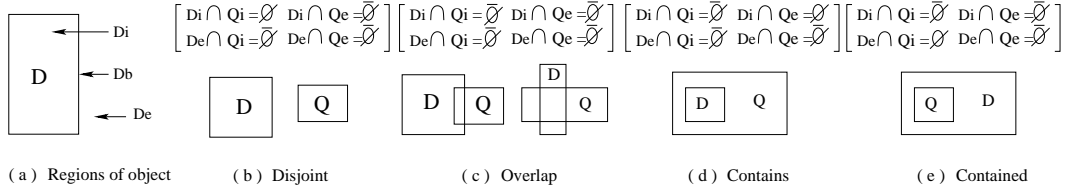


Figure 1: 4 Spatial Relations between two Objects

first applied by (Beigel & Tanin 1998) to build a spatial histogram for a set  $S$  of isothetic rectangles; the corresponding histogram is called Euler Histogram in (Sun et al. 2002). To build an Euler histogram  $H$  for  $S$ , the isothetic MBR containing the whole  $S$  is first divided into  $n_1 \times n_2$  equal cells; for instance, figure 2(a) illustrates the  $5 \times 5$  equal cells which is also called the  $5 \times 5$  grid. The histogram  $H$  is therefore referred to the histogram on the  $n_1 \times n_2$  grid; and the  $n_1 \times n_2$  grid is the *grid* of  $H$ . A rectangle *aligns with* the grid of  $H$  if its 4 edges align with the 4 lines in the grid (see Figure 2(b) for example). A rectangle *occupies* the  $w \times h$  cells if its horizontal edges cross  $w$  cells and its vertical edges cross  $h$  cells (see Figure 2(c) for example).

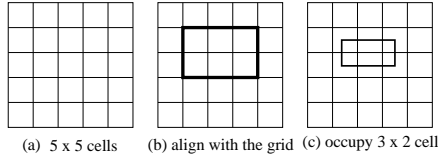


Figure 2: Histogram Grid

To build the Euler histogram on the  $n_1 \times n_2$  grid, the  $(2n_1 - 1) \times (2n_2 - 1)$  buckets are needed to be allocated to keep the information, which correspond to the internal edges, cells, or nodes, where each bucket in the histogram stores an integer, such that:

- The integer corresponding to a cell in the grid is increased by 1 if an object intersects the cell.
- The integer corresponding to a node in the grid is increased by 1 if an object contains the node.
- The integer corresponding to an edge in the grid is decreased by 1 if the edge crosses an object.

Figure 3(a) gives an example of Euler histogram. Clearly, if we sum up all the buckets one object covers, the result should be 1 according to the Euler theorem. However, if an object with a hole inside, the summation is 0 (shown in Figure 3(b)).

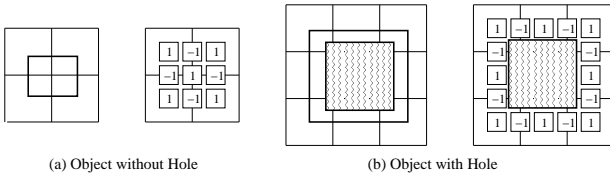


Figure 3: Euler Histogram

Based on the Euler histogram techniques, very recently the four relations, *disjoint*, *overlap*, *contains*, and *contained* between a query rectangle and an object in  $S$ , have been investigated in (Sun et al. 2002).

It has been shown that the information in one Euler histogram is not enough to determine the break-down information about the number of objects in  $S$  that fall into these 4 relation categories, respectively. For instance, in Figure 4 the two different scenarios (Figure 4(a) and Figure 4(b)) lead to the same histogram (Figure 4(c)). However, the break-down information against the same query rectangle (the shaded rectangle in Figures 4(a) and 4(b)) is different with respect to these two different scenarios; and thus, it is impossible to get the exact break-down information by one Euler histogram.

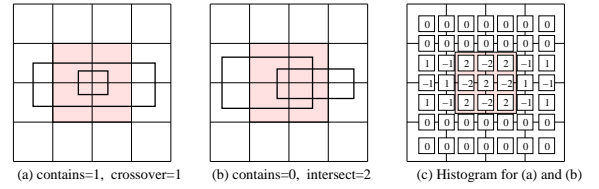


Figure 4: A Counter Example

As illustrated in Figure 1(c), we separate the overlap relation into 1) *one-end intersect* (the left figure in Figure 1(c)), and 2) *cross-over* (the right figure in Figure 1(c)) because of their different behaviors in Euler histogram.

Suppose that  $H$  is a  $n_1 \times n_2$  Euler histogram for a set  $S$  of objects (rectangles), and  $Q$  is a query rectangle aligning with the  $n_1 \times n_2$  grid.

- $|S|$  denotes the total number of objects in  $S$
- $N_{cs}$  denotes the number of objects in  $S$  which  $Q$  contains (shown in Figure 5(a), the object  $a$ ).
- $N_{it}$  denotes the number of objects in  $S$  which (one-end) intersect  $Q$  (shown in Figure 5(a), the object  $b$ ).
- $N_{cr}$  denotes the number of objects in  $S$  which cross over  $Q$  (shown in Figure 5(a), the object  $c$ ).
- $N_{ds}$  denotes the number of objects in  $S$  which disjoint with  $Q$  (shown in Figure 5(a), the object  $d$ ).
- $N_{cd}$  denotes the number of objects in  $S$  by which  $Q$  is contained (shown in Figure 1(e))
- $P_i$  denotes the number of objects in  $S$  whose interiors intersect the interior of  $Q$ . (the dark shadowed part in Figure 5(b))
- $P_e$  denotes the number of objects in  $S$  whose interiors intersect the exterior of  $Q$ . (the light shadowed part in Figure 5(b))

Based on the facts shown in Figure 3 and Figure 5, the following three equations are immediate.

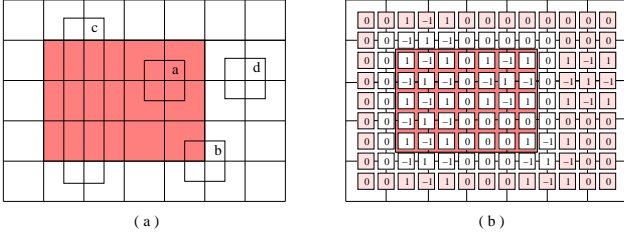


Figure 5: Compute  $P_i$  and  $P_e$  for *Disjoint, One-end intersect, Cross-over, Contains Relation*

$$|S| = N_{cs} + N_{it} + N_{cr} + N_{cd} + N_{ds} \quad (1)$$

$$P_i = N_{cs} + N_{it} + N_{cr} + N_{cd} \quad (2)$$

$$P_e = N_{it} + 2N_{cr} + N_{ds} \quad (3)$$

Note that  $P_i$  and  $P_e$  can be obtained by adding up all the integers contained in the corresponding regions, respectively; that is, the region for  $P_i$  is the interior of  $Q$  and the region for  $P_e$  is the exterior of  $Q$ . Further, we have to count a cross-over object twice in (3). For example, in Figure 5,  $P_i = 3$  that means there are 3 objects which interiors intersect the interior of  $Q$ , whereas  $P_e = 4$  that means there are 4 objects which interiors intersect the exterior of  $Q$ . But note here, because the object c is a cross-over object which contributes 2 to  $P_e$ , in fact there are only 3 objects (the object b,c and d) which interiors intersect the exterior of  $Q$ .

For the object by which  $Q$  is contained (shown in Figure 1(e)), it contributes 1 to  $P_i$  but contributes 0 to  $P_e$  (shown in Figure 3(b)).

In the equations (1) - (3), there are 5 variables to be fixed. Actually, it tends to be impossible to create more equations without introducing new variables. Therefore, 3 approximate algorithms have been proposed (Sun et al. 2002) to find the approximate solutions for these 4 variables excluding  $N_{ds}$  that can be always computed exactly. In the first approximate algorithm,  $N_{cd}$  and  $N_{cr}$  have been both ignored; and thus, three equations are just enough for fixing the three remaining variables. In the second approximate algorithm,  $N_{cr}$  is also omitted, while certain one-end intersecting objects are also omitted in order to introduce a new equation. In the third algorithm, the first and second approximate algorithms are used alternately to deal with a multi-resolution Euler histogram; that is, the Euler histogram is decomposed into a set of Euler sub-histograms according to the object areas.

### Motivation

In some applications, the cross-over relation is not always negligible. On the other hand, it should be clear that the three equations (1) - (3) can support only 3 variables. Consequently, it would be ideal if in a histogram, there are only 3 relations.

This is the motivation for us to develop a new multi-resolution algorithm. Further, the trade-off between the limited storage space and the accuracy is another motivation for us to investigate a new approximate algorithm.

### 3 Multi-resolution Algorithm

In this section, we will present two multi-resolution algorithms. The first algorithm will provide an exact answer to the five variables in the equations (1) - (3),

while the second algorithm will provide an approximate answer regarding a given storage space. We start with the presentation of the exact algorithm.

#### 3.1 An Exact Multi-resolution Algorithm

The basic idea of the algorithm is based on the above motivation. We decompose the Euler histogram on the  $n_1 \times n_2$  grid into a set  $A$  of Euler histograms on the  $n_1 \times n_2$  grid, such that the objects in each histogram in  $A$  have only three relations to a given query rectangle.

Let  $Q_{i \times j}$  denote a rectangle aligning with the  $n_1 \times n_2$  grid and occupying  $i \times j$  cells. Suppose that  $H_{w \times h}$  denotes the  $n_1 \times n_2$  Euler histogram for a set  $S_{w \times h}$  of objects (rectangles), where each object in  $S_{w \times h}$  occupies  $w \times h$  cells in the  $n_1 \times n_2$  grid. We have the following theorem.

**Theorem 1:** Each  $H_{w \times h}$  can provide an exact solution to identify  $N_{ds}$ ,  $N_{it}$ ,  $N_{cs}$ ,  $N_{cd}$  and  $N_{cr}$  for a given  $Q_{i \times j}$ .

**Proof:** When a  $Q_{i \times j}$  is used to query an Euler histogram  $H_{w \times h}$ , there are three cases by comparing  $w \times h$  with  $i \times j$ :

- **Case 1**  $w \leq i$  and  $h \leq j$  (shown in Figure 6(a)).
- **Case 2**  $w > i$  and  $h > j$  (shown in Figure 6(b)).
- **Case 3** ( $w > i$  and  $h \leq j$ ) or ( $w \leq i$  and  $h > j$ ) (shown in Figure 6(c)).

Below we prove the theorem with respect to these three cases.

Note that in case 1, the width of any object in  $S_{w \times h}$  is not greater than that of  $Q_{i \times j}$ , nor the height is. It can be immediately verified that no object in this histogram can have cross-over relation with  $Q$ , neither the contained relation. Consequently,  $N_{cr} = 0$  and  $N_{cd} = 0$ . Clearly, the remaining three variables  $N_{cs}$ ,  $N_{it}$ , and  $N_{ds}$  can be fixed from the equations (1) - (3).

For a similar reason, we can immediately show that in case 2, there is no cross-over relation nor contains relation; that is,  $N_{cr} = 0$  and  $N_{cs} = 0$ . Thus, the remaining three variables can also be fixed by the three equations.

In case 3, it is also immediate that there is no contained relation nor contains relation; that is,  $N_{cd} = 0$  and  $N_{cs} = 0$ . Again, the three remaining variables can be fixed by the three equations.  $\square$

Our exact algorithm is based on Theorem 1. It can be described below.

#### Exact Multi-resolution Algorithm

**Step 1** Scan the dataset  $S$  and allocate each object into a corresponding  $S_{w \times h}$ .

**Step 2** Construct the Euler histogram  $H_{w \times h}$  for each  $S_{w \times h}$  on the  $n_1 \times n_2$  grid.

**Step 3** With respect to each  $H_{w \times h}$  and a given  $Q_{i \times j}$ , the exact results of  $N_{cs}$ ,  $N_{cd}$ ,  $N_{cr}$ ,  $N_{ds}$  and  $N_{it}$  can be computed by the methods shown in the proof of Theorem 1; adding them up respectively to get the global  $N_{cs}$ ,  $N_{cd}$ ,  $N_{cr}$ ,  $N_{ds}$ , and  $N_{it}$ .

#### Storage Space

An Euler histogram on the  $n_1 \times n_2$  grid requires  $O(n_1 \times n_2)$  storage space. In the worst case, our algorithm may need  $O(n_1 \times n_2)$  Euler histograms. Consequently, our algorithm requires  $O(n_1^2 \times n_2^2)$  storage space; however, in practice the actual storage space

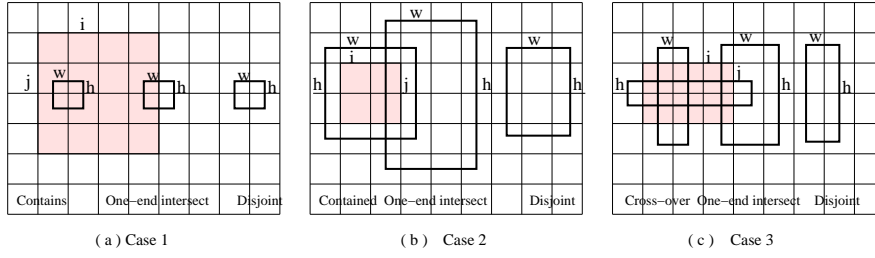


Figure 6: Three Cases by Comparison between  $Q_{i \times j}$  and  $H_{w \times h}$

may be much smaller than that in the worst case. For instance, if we build the multi-resolution histograms for the California road segments extracted from the US Census Tiger (TIGER 2000) on the  $360 \times 180$  grid, there are only 37 histograms instead of  $360 \times 180$  histograms needed.

Further, it can be immediately shown that the objects falling into  $S_{1 \times 1}$ ,  $S_{1 \times 2}$ ,  $S_{2 \times 1}$  or  $S_{2 \times 2}$ , may be put into one Euler histogram instead of 4 histograms.

### Query Processing Costs

For every histogram  $H_{w \times h}$ , a  $Q_{i \times j}$  can be answered in a constant time by using *prefix-sum* techniques. The interested reader will find the details of this techniques from (Ho, Agrawal, Megiddo & Srikant 1997).

### 3.2 Hybrid Multi-resolution Approximate Algorithm (HMA)

Clearly, our exact multi-resolution algorithm may be very space consuming for some applications; it may exceed a given storage space. In this subsection, we provide a multi-resolution approximation algorithm restricted to a given storage space.

Suppose that a fixed storage space is given, say, only  $k$  histograms on the  $n_1 \times n_2$  grid are allowed. The main idea of our algorithm is to construct the first  $k - 1$  histograms which can provide the exact solutions, while the remaining objects are all dumped into the  $k$ th histogram which will give an approximate solution only. Intuitively, less objects fall into the  $k$ th histogram, more accuracy of the approximation may be globally expected in general. Therefore, in our algorithm we first choose  $k - 1$   $S_{w_i \times h_i}$ s such that  $\sum_{i=1}^{k-1} |S_{w_i \times h_i}|$  is maximized. Our approximate algorithm (HMA) may be presented as follows.

#### HMA

**Step 1** Scan  $S$  to allocate each object into an appropriate  $S_{w \times h}$ . Choose  $k - 1$   $S_{w_i \times h_i}$ s such that  $\sum_{i=1}^{k-1} |S_{w_i \times h_i}|$  is maximized.

**Step 2** Built the Euler histogram  $H_{w_i \times h_i}$  on the  $n_1 \times n_2$  grid for each chosen  $S_{w_i \times h_i}$ . Calculate the exact answers for each  $H_{w_i \times h_i}$ .

**Step 3** Create the Euler histogram  $H_{last}$  for the remaining objects (rectangles). Use the following approximate algorithm to calculate the answers for a given  $Q_{i \times j}$ .

**Step 4** Sum up the results respectively in steps 2 and 3.

Obviously, we can use the first or second approximate algorithm in (Sun et al. 2002) to approach  $H_{last}$  (Step 3). However, it should be clear that such a choice is not appropriate since  $H_{last}$  may contain the

objects with various different sizes. Below we present a hybrid algorithm for Step 3.

### A Hybrid Algorithm

The last histogram  $H_{last}$  collects all the objects from the remaining set  $S_{last}$  of objects. Suppose we keep the  $n_1 \times n_2$  summation numbers  $s_{w,h}$  (for  $1 \leq w \leq n_1$ ,  $1 \leq h \leq n_2$ ), such that each  $s_{w,h}$  represents the number of objects in  $S_{last}$  which occupy  $w \times h$  cells in the grid. For a given  $Q_{i,j}$ , we can group the objects in  $S_{last}$  into 3 groups.

Group 1: Any object in this group occupies  $w \times h$  cells, such that  $w \leq i$  and  $h \leq j$ . By Theorem 1, the only possible relations corresponding to  $Q_{i \times j}$  are: *contains*, *one-end intersect*, *disjoint*.

Group 2: Any object in this group occupies  $w \times h$  cells where  $w > i$  and  $h > j$ . By Theorem 1, the only possible relations corresponding to  $Q_{i \times j}$  are: *contained*, *one-end intersect*, *disjoint*.

Group 3: Any object in this group occupies  $w \times h$  cells with  $w > i$  and  $h \leq j$ , or with  $w \leq i$  and  $h > j$ . By Theorem 1, the only possible relations corresponding to  $Q_{i \times j}$  are: *crossover*, *one-end intersect*, *disjoint*.

Note that each of these three groups includes the two relations: (one-end) intersect and disjoint. If we assume that the objects in  $S_{last}$  are distributed evenly over the grid, then we can use  $\frac{u}{u+v+x} \times (N_{it} + N_{ds})$ ,  $\frac{v}{u+v+x} \times (N_{it} + N_{ds})$ , and  $\frac{x}{u+v+x} \times (N_{it} + N_{ds})$  respectively to represent the total number of one-end intersecting objects and disjoint objects in each group. Here,  $u$ ,  $v$ , and  $x$  denote the total number of objects in group 1, group 2 and group 3, respectively. Note  $u$ ,  $v$  and  $x$  can be computed from  $\{s_{w,h}\}$ . These give us the following equations for a given  $Q_{i \times j}$ :

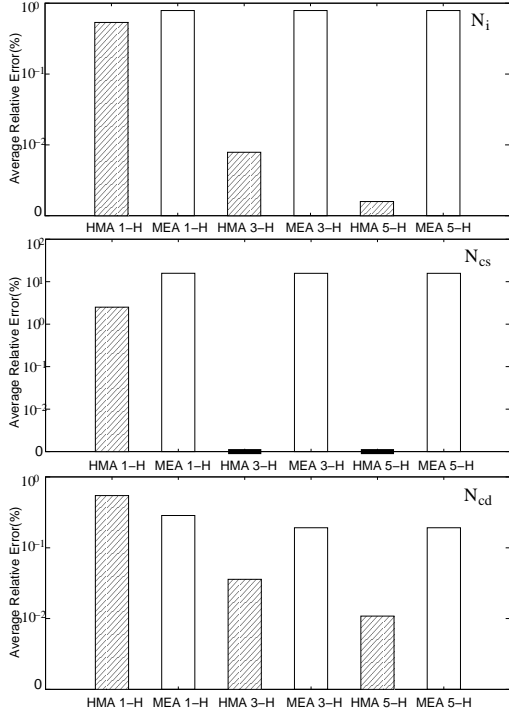
$$\frac{N_{cs} + \frac{u}{u+v+x} \times (N_{it} + N_{ds})}{N_{cr} + \frac{v}{u+v+x} \times (N_{it} + N_{ds})} = \frac{u}{v} \quad (4)$$

$$\frac{N_{cd} + \frac{x}{u+v+x} \times (N_{it} + N_{ds})}{N_{cr} + \frac{v}{u+v+x} \times (N_{it} + N_{ds})} = \frac{x}{v} \quad (5)$$

Combine these 2 equations with the equations (1),(2),(3), we obtain  $N_{cd}$ ,  $N_{cs}$ ,  $N_{it}$ ,  $N_{cr}$  and  $N_{ds}$  for  $H_{last}$ .

### HMA Computation Cost

The selection of  $k - 1$   $S_{w \times h}$ s can easily run in  $O(\min(k \times n_1 \times n_2, (\log(n_1) + \log(n_2)) \times n_1 \times n_2))$  time. The prefix-sum techniques (Ho et al. 1997) can also be adopted to make the computation of  $u$ ,  $v$  and  $x$  run in a constant time.



$T_{1 \times 1}$		1-H	3-H	5-H
$N_i$	HMA	0.535707	0.007844	0.001581
	MEA	0.78796	0.78796	0.78796
$N_{cs}$	HMA	2.51	0	0
	MEA	15.8	15.7	15.7
$N_{cd}$	HMA	0.546296	0.035803	0.010803
	MEA	0.285494	0.191358	0.191358

Figure 7: Experiment Results for  $T_{1 \times 1}$  Regarding Different Storage Spaces

#### 4 Performance Evaluation

In this section we evaluate the performance of our new algorithm HMA. Specifically, we will evaluate the accuracy of HMA comparing with the techniques in (Sun et al. 2002) but not the efficiency; this is because in HMA, querying one histogram runs in a constant time as the algorithms in (Sun et al. 2002) do.

##### Dataset

In our initial experiment, we use a real dataset California Road segments selected from the US Census TIGER (TIGER 2000). It consists of 2,837,688 objects. We normalize the dataset into a given  $360 \times 180$  grid.

##### Query Sets

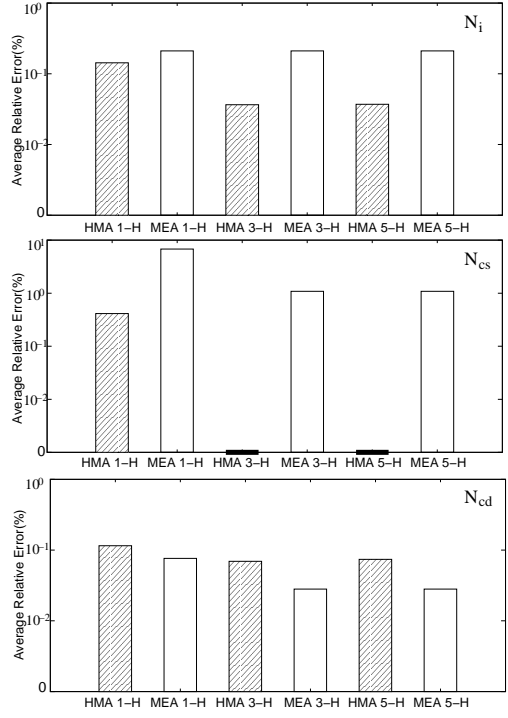
In our experiments, we use 4 sets of query rectangles. These include different shapes such as square rectangles as well as long and narrow rectangles as described below.

$$T = \{T_{1 \times 1}, T_{2 \times 2}, T_{5 \times 2}, T_{10 \times 2}\}$$

where each  $T_{i \times j}$  consists of  $(360 - i + 1) \times (180 - j + 1)$  query rectangles in the grid, each of which occupies  $i \times j$  cells.

##### Error Metrics

As mentioned earlier, we evaluate the approximation accuracy of our algorithm by using the average



$T_{2 \times 2}$		1-H	3-H	5-H
$N_i$	HMA	0.142695	0.036551	0.037129
	MEA	0.210558	0.210558	0.210558
$N_{cs}$	HMA	0.413229	0	0
	MEA	6.80887	1.08661	1.08661
$N_{cd}$	HMA	0.115155	0.069249	0.073917
	MEA	0.076252	0.028011	0.028011

Figure 8: Experiment Results for  $T_{2 \times 2}$  Regarding Different Storage Spaces

relative error below in our performance study.

$$\frac{\sum_{q \in T_{i \times j}} \epsilon_q}{|T_{i \times j}|} \quad (6)$$

where

$$\epsilon_q = \begin{cases} \frac{|e - e'|}{e} & \text{if } e \neq 0 \\ |e - e'| & \text{otherwise} \end{cases}$$

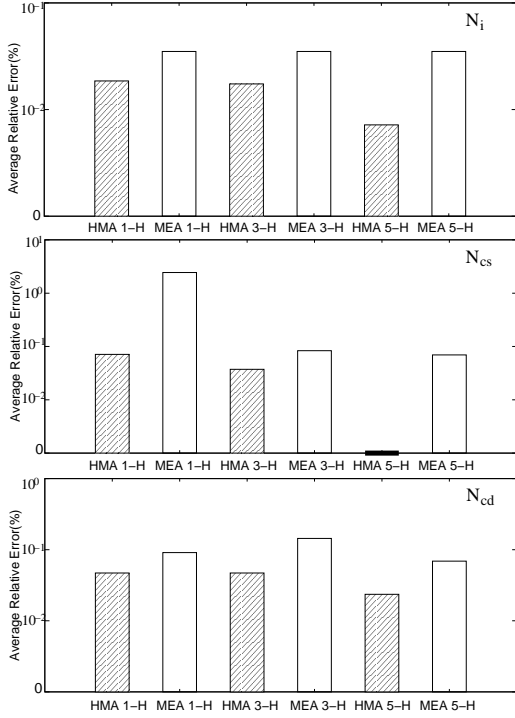
Here,  $e$  is an exact value and  $e'$  is an approximate value.  $\epsilon_q$  is the average relative error of a single query  $q$ .

##### Implementation

We examined the performance of HMA against 3 different storage space requirements, 1 histogram, 3 histograms, and 5 histograms. For comparison reason, S-Euler, AproxEuler and M-Euler (Sun et al. 2002) are also implemented for different query rectangles. We name such a collection of these three algorithms as MEA.

In our experiments, we record the average relative errors for a given storage space and a given query set for HMA and MEA, respectively. We denote:

- the number of objects, overlapping a given query rectangle, by  $N_i$  ( $N_i = N_{it} + N_{cr}$ );
- the number of objects, which a given query rectangle contains, by  $N_{cs}$ ;



$T_{5 \times 2}$		1-H	3-H	5-H
$N_i$	HMA	0.018543	0.017415	0.007184
	MEA	0.035061	0.035061	0.035061
$N_{cs}$	HMA	0.071172	0.037364	0
	MEA	2.43689	0.083526	0.069447
$N_{cd}$	HMA	0.047078	0.047078	0.023539
	MEA	0.091018	0.144373	0.069048

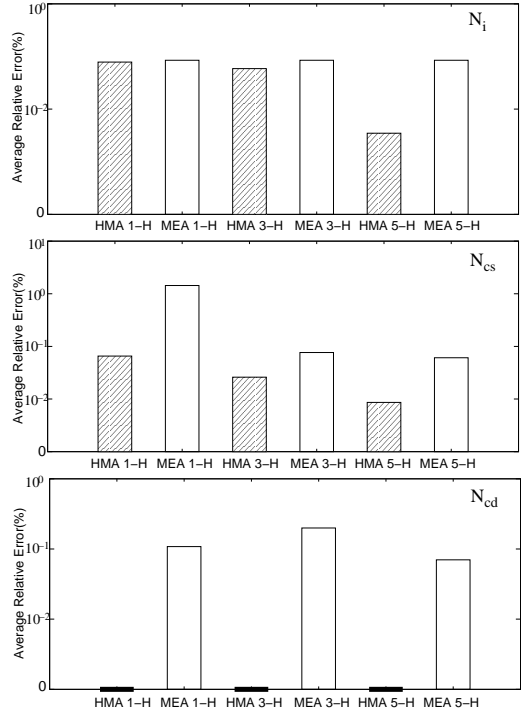
Figure 9: Experiment Results for  $T_{5 \times 2}$  Regarding Different Storage Spaces

- the number of objects, by which a given query rectangle is contained, by  $N_{cd}$ .

We did not evaluate the solutions for the number of objects ( $N_{ds}$ ) that disjoint with a given query rectangle in our experiments, because both HMA and MEA can provide the exact answers to this problem.

In our experiments, we found that HMA provides a better answer (i.e., with smaller relative errors) than MEA does; this tends to be more significant when a query object is small or long and narrow. Recall that we present our experiment results regarding some small query rectangles or long and narrow query rectangles. We use 1-H to represent the storage limitation when only 1 histogram is used, 3-H and 5-H to represent the situation where only 3 histograms and 5 histograms can be used respectively. Note that in HMA, the last histogram needs  $n_1 \times n_2$  space to store the statistic information. Therefore, the storage space for the last histogram in HMA needs about one fourth space more than the MEA does. However, our experiment results suggest that the additional storage space may bring enormous benefits. We can see that in our experiment results, HMA with  $i$  histograms even out-performs the MEA with  $i + 2$  histograms (for  $1 \leq i \leq 3$ ).

Figure 7 and Figure 8 show the experiment results for  $T_{1 \times 1}$  and  $T_{2 \times 2}$ . Note that in California Road dataset, the number of small objects is much greater than the number of the large objects; consequently  $N_{cs}$  may be quite large while  $N_{cd}$  could be negligible. In fact, in our experiments we found that over all queries in  $T_{1 \times 1}$  ( $T_{2 \times 2}$ ), the total  $N_{cs}$  is about 2.5 mil-



$T_{10 \times 2}$		1-H	3-H	5-H
$N_i$	HMA	0.027952	0.024267	0.005901
	MEA	0.029108	0.029108	0.029108
$N_{cs}$	HMA	0.06552	0.026055	0.008623
	MEA	1.43437	0.07646	0.060935
$N_{cd}$	HMA	0	0	0
	MEA	0.10823	0.198953	0.070031

Figure 10: Experiment Results for  $T_{10 \times 2}$  Regarding Different Storage Spaces

lions (10.7 millions), while the total  $N_{cd}$  is 398 (80). Therefore, the estimation accuracy of  $N_{cs}$  is much more important than that of  $N_{cd}$ . Figure 7 and Figure 8 show that the HMA provides significantly better estimation of  $N_{cs}$  than MEA does. Though the estimation of  $N_{cd}$  in HMA is slightly worse than that in MEA for testing  $T_{2 \times 2}$ , this will not bring a significant impact on the performance of HMA due to the 2 reasons: 1) the difference between is not significant and both of them are within 0.1%, 2) the total number of  $N_{cd}$  over all the queries is very small 80 (less than about 0.001 per query on average).

In MEA, the cross-over objects have been ignored. The experiment results in Figure 9 and Figure 10 show the approximate error may be propagated to the other parameters because of this kind of ignoring. When a query rectangle is narrow and long, cross-over objects should not be ignored. In fact, our algorithm HMA provides a much more accurate results than MEA for  $T_{5 \times 2}$  and  $T_{10 \times 2}$ , as depicted in Figures 9 and 10.

## 5 Conclusion and Remarks

In this paper, we investigated the problem of effectively obtaining a preview of spatial query processing results using spatial histograms. We first present an exact algorithm based on a multi-resolution paradigm. To conform to a given storage space, we also provide a hybrid multi-resolution algorithm by combining the geometric information with the statistic information. Our experiments suggest that our

techniques, developed in this paper, out-perform the existing techniques.

As a possible future study, we will investigate the randomized algorithmic techniques to approach the problem in this paper.

## References

- Aboulnaga, A. & Naughton, J. F. (2000), Accurate estimation of the cost of spatial selections, *in* 'ICDE'00, Proceedings of the 16th International Conference on Data Engineering', pp. 123–134.
- Acharya, S., Poosala, V. & Ranmaswamy, S. (1999), Selectivity estimation in spatial databases, *in* 'SIGMOD'99, Proceedings ACM SIGMOD International Conference on Management of Data', pp. 13–24.
- Beigel, R. & Tanin, E. (1998), The geometry of browsing, *in* 'Proceedings of the Latin American symposium on Theoretical Informatics, 1998, Brazil', pp. 331–340.
- Egenhofer, M. J. & Herring, J. R. (1994), Categorizing binary topological relations between regions, lines, and points in geographic databases, *in* M. J. Egenhofer, D. M. Mark & J. R. Herring, eds, 'The 9-Intersection: Formalism and Its Use for Natural-Language Spatial Predicates. National Center for Geographic Information and Analysis, Report 94-1', pp. 13–17.
- Garofalakis, M., Gehrke, J. & Rastogi, R. (2002), Query and mining data streams: You only get one look, *in* 'VLDB'02, Proceedings of the 25th International Conference on Very Large Data Bases: Tutorial'.
- Harary, F. (1969), *Graph Theory*, Addison-Wesley Publishing Company.
- Ho, C. T., Agrawal, R., Megiddo, N. & Srikant, R. (1997), Range queries in olap data cubes, *in* 'SIGMOD'97, Proceedings ACM SIGMOD International Conference on Management of Data', pp. 73–88.
- Jin, J., An, N. & Sivasubramaniam, A. (2000), Analyzing range queries on spatial data, *in* 'ICDE'00, Proceedings of the 16th International Conference on Data Engineering', pp. 525–534.
- Lipton, R. J., Naughton, J. F. & Schneider, D. A. (1990), Practical selectivity estimation through adaptive sampling, *in* 'SIGMOD'90, Proceedings ACM SIGMOD International Conference on Management of Data', pp. 1–11.
- Sun, C., Agrawal, D. & Abbadi, A. E. (2002), Exploring spatial datasets with histograms, *in* 'ICDE'02, Proceedings of the 18th International Conference on Data Engineering', pp. 93–102.
- TIGER (2000), Tiger/line files, Technical report, U.S. Census Bureau, Washington, DC.
- Xiao, J., Zhang, Y. & Jia, X. (2001), 'Clustering non-uniform-sized spatial objects to reduce i/o cost for spatial-join processing', *The Computer Journal* **44**(5), 384–397.
- Zhou, X., Truffet, D. & Han, J. (1999), Efficient polygon amalgamation methods for spatial olap and spatial data mining, *in* 'SSD'99, Proceedings of the 6th International Symposium', pp. 167–187.