Summarizing Spatial Relations - A Hybrid Histogram*

Qing Liu[1,2], Xuemin Lin[1,2], and Yidong Yuan[1,2]

[1] University of New South Wales, Australia
[2] National ICT Australia
{qingl, lxue, yyidong}@cse.unsw.edu.au

Abstract. Summarizing topological relations is fundamental to many spatial applications including spatial query optimization. In this paper, we examine the selectivity estimation for range window query to summarize the four important topological relations: contains, contained, overlap, and disjoint. We propose a novel *hybrid histogram* method which uses the concept of Min-skew partition in conjunction with Euler histogram approach. It can effectively model object spatial distribution. Our extensive experiments against both synthetic and real world datasets demonstrated that our hybrid histogram techniques improve the accuracy of the existing techniques by about one order of magnitude while retaining the cost efficiency.

1 Introduction

For range query in the spatial databases, users are looking for data objects whose geometries lie inside or overlap a query window. In many applications, however, users are more interested in summarized information instead of objects' individual properties . Especially with the availability of a huge collection of on-line spatial data [1, 2, 3] (e.g. large digital libraries/archives), it becomes extremely important to support *interactive* queries by *query preview* [4, 1]. These applications require systems to provide a fast summarized spatial characteristics information. Summarizing spatial datasets is also a key to spatial query process optimization.

In this paper, we investigate the selectivity estimation problem of summarizing topological relations between rectangular objects and window query. Several techniques have been proposed for estimating the selectivity [5, 6]. Technique based on histogram to approximate data distributions is widely used by current database systems [7]. Histogram-based techniques can be classified into two categories: 1) *data partition techniques* and 2) *cell density techniques* [8]. The Min-skew algorithm [9] and the SQ-histogram technique [10] belong to the first category. They group "similar" objects together into one bucket for estimating the number of *disjoint* and *non-disjoint* objects with respect to window query.

* This project was supported by ARC Discovery Grant DP0346004.

Y. Zhang et al. (Eds.): APWeb 2005, LNCS 3399, pp. 464–476, 2005.

Techniques based on cell density [4,11,3,8] propose to divide the object space evenly into a number of disjoint cells, and record object density information in each cell. Cumulative density based approach [11] and Euler histogram [4] can provide the exact solutions against the *aligned* window query (to be defined in Section 2) for non-disjoint and disjoint topological relations only. Euler-Approx [3] and Multiscale Histogram [8] substantially extended the Euler histogram techniques for summarizing the 4 important binary topological relations: "*contains*", "*contained*", "*overlap*", and "*disjoint*" (to be defined in Section 2) against the aligned window query.

In this paper, we will focus on these 4 relations against aligned window query. Specifically, we developed a novel hybrid histogram method that combines Min-skew technique with Euler histogram approach. By combining these two techniques together, this hybrid histogram may lead to more accurate solution for aligned window query. We evaluate our new techniques by both synthetic and real world datasets. Our experiment results demonstrated that the hybrid histogram may improve the accuracy of the existing techniques by about one order of magnitude while retaining the cost efficiency.

The rest of the paper is organized as follows. In Section 2, we provide preliminaries and related work. Section 3 presents our hybrid histogram construction algorithm, query algorithm and analysis of this structure. Section 4 evaluates the proposed methods through extensive experiments with synthetic and real datasets, and Section 5 concludes the paper with direction for future work.

2 Preliminary

In this section, we give a brief overview of Min-skew algorithm [9], Euler histogram [4], Euler-Approx algorithm [3] and Multiscale Histogram [8]. These techniques are closely related to our work in this paper. First we introduce the middle-resolution topological relations.

A binary topological relation between two objects, P and Q, is based upon the comparison of P's *interior*, *boundary*, *exterior* with Q's interior, boundary,

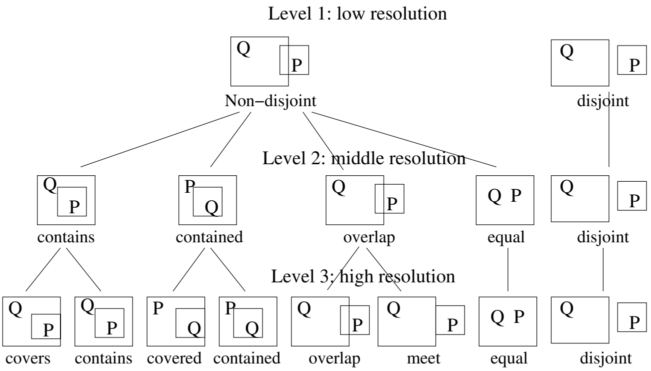


Fig. 1. Topological Relations between Two Objects

and exterior [12]. It can be classified into 8 high-resolution topological relations according to the 9-intersection model [12], and can be also classified into the middle-resolution topological relations by removing the intersections involving the object boundaries [13, 3] (Figure 1).

Aligned window query means the query window contains whole cells but not part of the cells.

2.1 Min-Skew Algorithm

Acharya, Poosala, and Ramaswamy proposed a partitioning scheme to build histograms for spatial data called Min-skew partition [9]. It is regarded as the winner among several other histogram techniques. The technique uses a uniform grid that covers the whole space and their spatial densities as input. Initially one bucket represents the whole space. The algorithm iteratively splits a bucket into two until the histogram has the required number of buckets B. The algorithm tries to minimize the spatial-skew, defined as the variance of the number of objects in the grid cells constituting the bucket, at each step.

2.2 Euler Histogram

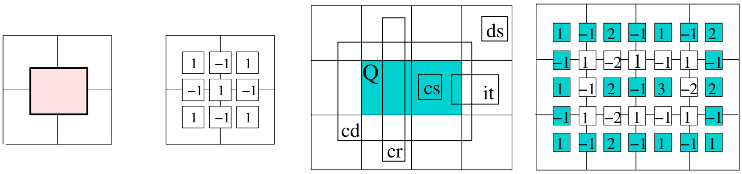
To construct an Euler histogram [4], the whole space is first divided evenly into $n_1 \times n_2$ disjoint cells. For each node, edge and cell, a bucket would be allocated respectively. So the total space required is $(2n_1 - 1) \times (2n_2 - 1)$. For every object insertion, an update is needed for all the nodes, edges and cells that the object intersects: the value of relevant cell and node is increased by 1 and the value of relevant edge is decreased by 1. Figure 2(a) gives an example of an Euler histogram for a dataset with only one object.

Table 1 lists the symbols that will be used frequently throughout the paper. Given Q, we can get P_i by summing up all the bucket values inside Q. By Euler formula, we have:

$$N_{nds} = P_i \tag{1}$$

$$N_{nds} + N_{ds} = |S| \tag{2}$$

Equation (1) and equation (2) yield the exact solution for low-resolution relations N_{nds} and N_{ds}. In Figure 2(b) for example, given Q (shadow area), $N_{nds} = P_i = 2 - 1 + 3 = 4$, $N_{ds} = |S| - N_{nds} = 5 - 4 = 1$, which means there are 4 objects (cs, cd, it, cr) non-disjoint with Q and 1 object (ds) disjoints with Q.



(a) Euler histogram for one object (b) Compute Pi and Pe

Fig. 2. Euler Histogram and Query Method

Table 1. Frequent Symbols

Q	aligned window query
$\|S\|$	the total number of objects in dataset
P_i	the summation of all the bucket values inside Q
P_e	the summation of all the bucket values outside Q
N_{ds}	the number of objects that disjoint with Q
N_{nds}	the number of objects that non-disjoint with Q
N_{cs}	the number of objects that Q contains
N_{cd}	the number of objects by which Q is contained
N_{ov}	the number of objects that overlap Q
N_{it}	the number of objects that intersect Q
N_{cr}	the number of objects that crossover Q

2.3 Euler-Approx Algorithm

Sun, Agrawal and El Abbadi [3] proposed to use the histogram information outside query window, P_e, to solve middle-resolution relations: contains, contained, overlap and disjoint. The equal relation is merged into contains relation. It is shown the overlap relation has to be separated into two classes: intersect relation and crossover relation. This is because an object with intersect relation contributes 1 to the outside of query and an object with crossover relation contributes 2 to the outside of query (see Figure 2(b) object *it* and *cr* for example). So we have to deal with crossover and intersect relation respectively, and then sum them up to get overlap relation. In the rest of paper, the 5 relations represent contains, contained, intersect, crossover and disjoint relations. Again, by Euler formula, we have:

$$N_{cs} + N_{it} + N_{cr} + N_{cd} = P_i \tag{3}$$

$$N_{ds} + N_{it} + 2 \times N_{cr} = P_e \tag{4}$$

$$N_{cs} + N_{it} + N_{cr} + N_{cd} + N_{ds} = |S| \tag{5}$$

For example, in Figure 2(b), $P_i = 4$. This 4 comes from the object *cs*, *cd*, *cr*, and *it*. We can also have $P_e = 4$. This 4 comes from the object *ds*, *it*, and *cr* which contributes 2 to P_e.

The information in one Euler histogram is not enough to determine all the above 5 relations. To solve these 5 relations with only 3 exact equations, Sun et. al proposed three query algorithms: Simple-Euler, Euler-Approximate and Multi-resolution Euler Approximate. All algorithms are based on assumptions $N_{cd} = 0$ and/or $N_{cr} = 0$ (see [3] for details).

2.4 Multiscale Histogram

Lin et. al proposed a multiscale framework [8] which can provide exact solution for many real applications. The framework contains two parts: exact algorithm ($MESA$) and approximate algorithm ($MAPA$). In $MESA$, it is proved that if all the objects involved in one Euler histogram have at most four adjacent scales $(w, h), (w + 1, h), (w, h + 1), (w + 1, h + 1)$, the exact results can be obtained. So objects with adjacent scales are grouped together and one Euler histogram is constructed for each group. When storage space is limited to k, MAPA will build $k - 1$ exact histograms and 1 approximate histogram.

For an Euler histogram with a resolution $n_1 \times n_2$, the storage space required is $O(n_1 \times n_2)$. Both Euler-Approx and Multiscale Histogram run in constant time with the *prefix-sum* technique [14].

3 EM Histogram

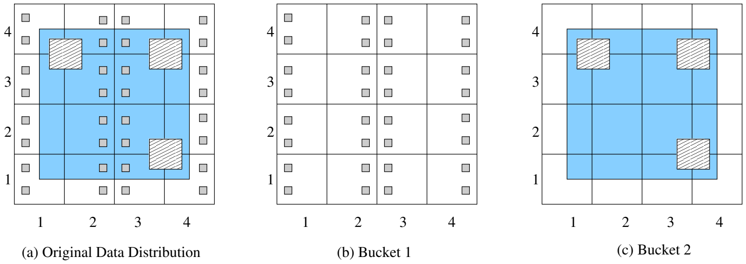
In this section, we introduce the hybrid histogram technique. The histogram captures information not only about the location of the data objects, but also about their sizes. These information are essential to the accuracy of 5 relations estimation. We call this hybrid histogram *EM histogram.* It takes advantages both from **E**uler histogram and **M**in-skew partition to achieve high accuracy as well as efficiency. We first identify our motivation.

Our study shows, despite different techniques have their own attractive aspects, they also have their own limited applicability due to this unique problem. Min-skew provides a good location estimation. But to approximate the objects within a bucket, all objects are presented by an average size object. Obviously this solution could not be applied for our problem because it is very unlikely all the objects in one bucket only have one relation with respect to a given query. Multiscale Histogram could provide a very accurate estimation if most of objects are involved in the exact histograms. But usually the histogram space is limited. If given 1 histogram space, no accurate result can be obtained for any objects. So estimation accuracy for using only 1 histogram is a critical component to solve the 5 relations problem. Motivated by these, next we present our EM histogram.

3.1 Histogram Construction

EM histogram includes two parts: Min-skew-like partition and Euler histogram. The construction method for Euler histogram is exactly the same as that in Section 2.2. For Min-skew-like partition, given a regular $n_1 \times n_2$ cells and object scale range, we partition the objects based on the location of their bottom-left corners as well as their scale information. So instead of only minimizing the location variance sum in Min-skew partition, Min-skew-like partition aims at minimizing the location as well as scale variance sum of all the buckets.

It generates B rectangular buckets whose edges are aligned with the cell boundaries. In each bucket, the distribution of object location and scale are almost uniform. Then the location uniform model is applied locally in each bucket. The scale uniform model could not be applied directly. But with this uniform model, the accuracy of estimation can be improved a lot in each bucket. Each bucket b records the following information:



(a) Original Data Distribution (b) Bucket 1 (c) Bucket 2

Fig. 3. Skewed Scale Distribution

- the spatial extent b_l.MBR $[(minX, maxX), (minY, maxY)]$ $(1 \leq l \leq B)$
- the scale matrix b_l.Matrix of the objects involved in the bucket $[(width_1, height_1)\ objNum_1, \ldots\ldots, (width_n, height_n)\ objNum_n]$ $(1 \leq l \leq B)$

Figure 3 gives an example for skewed scale data distribution with 2 buckets. The overall location distribution of this dataset is uniform. So by the Min-skew-like partition, 2 buckets are obtained. In bucket 1 (Figure 3(b)), b_1.MBR is [(1, 4), (1, 4)], b_1.Matrix is [(1, 1) 32]. In bucket 2 (Figure 3(c)), b_2.MBR is [(1, 4), (1, 4)], b_2.Matrix is [(2, 2) 3, (4, 4) 1].

3.2 Querying EM Histogram

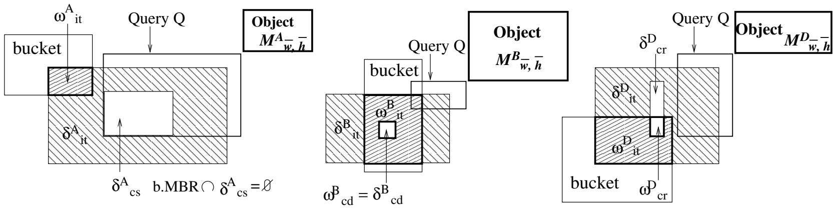
By Section 2.2, we have equation (3), (4) and (5) based on the Euler histogram. Next we will use the Min-skew-like partition to get 2 more equations to solve 5 relations: contains(cs), contained(cd), intersect(it), crossover(cr) and disjoint(ds).

Given a query Q and a bucket, we estimate its selectivity with respect to 5 relations based on the bucket scale matrix information. The basic idea is by the scales of objects and a given query, the objects in each bucket can be separated into 5 groups. In each group, a mean object will be calculated to represent the objects in that group. Then probabilistic method is applied to estimating the 5 relations. Details can be explained in 3 steps:

Step 1. For each bucket, Q divides the objects into five groups according to object scale $O_{w,h}$ and query scale $Q_{i,j}$. In each group, we know exactly how many objects may contribute to the specific relations.

Group A. $w \leq i$ and $h \leq j$ - at most 3 relations: cs, it and ds.
Group B. $w \geq i + 2$ and $h \geq j + 2$ - at most 3 relations: cd, it and ds.
Group C. $w \leq i$ and $h \geq j + 2$ - at most 3 relations: cr, it and ds.
Group D. $w \geq i + 2$ and $h \leq j$ - at most 3 relations: cr, it and ds.
Group E. $w = i + 1$ or $h = j + 1$ - at most 2 relations: it and ds.

In the example of Figure 3(c), there are 2 kinds of objects in this bucket. A query window with scale (3, 3) separates the scale matrix into five groups: A, B, C, D, E. Objects with (2, 2) scale belong to group A. So these objects will only contribute contains, intersect or disjoint relation to Q. No object belongs to group B, C, D which indicates no object contributes to contained



(a) Group A **(b) Group B** **(c) Group D**

Fig. 4. Possible Area regards to Different Relations in Group A, B, D

(crossover) relation. The objects with (4, 4) scale belong to group E. So they may only contribute to intersect and disjoint relation.

Step 2. Calculate a mean object $M^g_{\bar{w},\bar{h}}$ ($g \in \{A, B, C, D, E\}$) for each group. To make the computation more efficient, we use 5 mean objects $M^g_{\bar{w},\bar{h}}$ to represent all the objects in 5 groups respectively. The number of $M^g_{\bar{w},\bar{h}}$ (m_g) can be calculated by adding up all the objects with scales belonging to group g.

Step 3. Estimate the number of objects with respect to the 5 relations using probabilistic approach based on the mean object. Each mean object has 3 or 2 relations with respect to the query window $Q_{i,j}$. The occurring probabilities against the 3 or 2 relations can be calculated based on each mean object $M^g_{\bar{w},\bar{h}}$ by applying the location uniform model in each bucket.

In Figure 4, δs show the rectangular areas used by the mean object bottom-left corner regards to different relations for group A, B and D respectively. The rectangular areas for group C is similar to group D and for group E, it is similar to group A without the white area δ^A_{cs}. $\delta^g_{relation}$ ($g \in \{A, B, C, D, E\}$, $relation \in \{cs, cd, cr, it, ds\}$) denotes the possible area covered by the objects $M^g_{\bar{w},\bar{h}}$ that will contribute to relation *relation* in group g with respect to Q. For example, in Figure 4(a) δ^A_{cs} means if the mean object $M^A_{\bar{w},\bar{h}}$ has cs relation with Q, its bottom left corner should locate in this δ^A_{cs} area. So compared with other bucket estimation method, we would not compute the intersection area between bucket and query window, but between bucket and $\delta^g_{relation}$. Even there is no intersection between bucket and query window, the objects in this bucket may still contribute to some relations (eg. contained, crossover or intersect) to Q (see Figure 4(a, c) for example). N_{ds} can be computed directly from equation (3) and (5). We would not compute it using probability approach. There are 2 possible cases between $b.MBR$ and $\delta^g_{relation}$:

- case 1: $b.MBR \cap \delta^g_{relation} = \emptyset \quad (relation \in \{cs, cd, cr, it\})$
- case 2: $b.MBR \cap \delta^g_{relation} = \omega^g_{relation} \quad (relation \in \{cs, cd, cr, it\})$

In case 1, there is no object that will contribute to relation *relation* in this group. For example in Figure 4(a), $b.MBR \cap \delta^A_{cs} = \emptyset$, so all the objects in this bucket will not contribute to cs relation with respect to Q.

In case 2, the selectivity $\rho^g_{relation}$ that objects in the g group contribute to relation *relation* can be calculated by the ratio of $\omega^g_{relation}$ to the total area $b.MBR$. The number of objects which contribute to a specific relation in each group is $m_g \times \rho^g_{relation}$. The summations of the results from 5 groups with respect to 5 relations form the results of this bucket.

By summing up the estimation of different relations $(\alpha_{cr}, \beta_{it}, \gamma_{cs}, \mu_{cd})$ from each bucket, 2 more equations can be obtained:

$$N_{cs} : N_{cd} = \gamma_{cs} : \mu_{cd} \tag{6}$$

$$N_{cr} : N_{it} = \alpha_{cr} : \beta_{it} \tag{7}$$

Together with equation (3), (4) and (5), five relations $N_{cs}, N_{cd}, N_{cr}, N_{it}, N_{ds}$ can be solved.

3.3 EM Maintenance

If an object is updated, EM histogram may be also updated. First, for an insertion or deletion, the value of relevant node, cell and edge should be updated. Because Euler histogram is constructed in a cumulative fashion, an efficient update technique, Δ tree [15], can be applied. Second, based on the object's spatial location and scale, the scale matrix information in the corresponding bucket will also be updated.

3.4 EM Performance Analysis

The storage space required by Euler histogram is $(2n_1 - 1)(2n_2 - 1)$. And the space for buckets is Bn_1n_2 in the worst case (B is the number of buckets). The total actual storage space required by EM histogram can be calculated as $(2n_1 - 1)(2n_2 - 1) + Bn_1n_2$. But in practice, by our extensive experiments, it takes much smaller space than that in theory. This would be shown in our experiment part.

Because we can also represent the scale matrix information in each bucket by applying prefix-sum techniques, querying each bucket runs in constant time. The time for querying B buckets is $O(B)$. And we know an Euler histogram can be queried in constant time, so the total time for querying EM histogram is $O(B)$.

4 Performance Evaluation

This section experimentally evaluates the proposed methods. All the experiments were performed on Pentinum IV 1.80GHz CPU with 512 Mbytes memory.

The objective of this study is to show by taking advantages of Euler histogram and Min-skew-like techniques, EM histogram provides an accurate and efficient method for spatial selectivity estimation for middle-resolution topological relations, especially for non-uniform distribution dataset. In the first part of experiments, we will show the importance that we integrate Min-skew-like partition technique into our EM histogram. In the second part of experiment, we will show the importance of Euler histogram. And cost comparison will be examined in the third part. We evaluate the accuracy of the following techniques:

- EM Histogram
- Euler-Approx [3]
- Multiscale Histogram [8]
- Pure-Minskew Method: it is used to show the advantages of Euler histogram. In Pure-Minskew method, only bucket information is recorded without any Euler information. The spatial space and object scale space are partitioned by Min-skew algorithm. In each bucket, the bucket spatial extent as well as object scale matrix are maintained. For previous Min-skew algorithm, only 1 average scale information is recorded. Modified in this way, the Pure-Minskew method can also be used to answer the query for middle-resolution relations.

Datasets. In our experiment, both real-world and synthetic datasets are used. To do a fair comparison with Euler-Approx and Multiscale histogram regarding accuracy, we adopt the 360 × 180 resolution to evaluate the accuracy of our algorithms, as this resolution was used in [3] and [8] to provide the experiment results. The 360 × 180 grid is a simulation of the earth resolution by the longitude and latitude. Below are the datasets used.

- **Ca_road** consists of the 2, 851, 627 California road segments obtained from the US Census TIGER [16] dataset. We normalized the dataset into the 360 × 180 grid.
- **Zipf** is a synthetic dataset with one million square objects. Both the side length and the spatial location of the object follow a Zipf distribution.

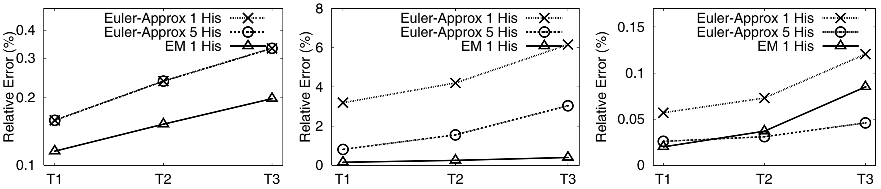
Query Sets. We adopt the same query setting in [8]. This is because this setting simulates various user query patterns. Query windows are divided into 2 classes, small and non-small. A query window in small class has a scale such that the width and height are both smaller than 5, while a query window in non-small class has either height between 6 and 20 or width between 6 to 20. We randomly generate 3 different sets of windows, T_1, T_2, and T_3, each of which has 100, 000 query windows.

In T_1, 20% of the 100, 000 query windows are in the small class. In T_2, 40% of the query windows are in the small class, while in T_3, 80% of the query windows are in the small class.

Error Metrics. We adopt average relative error for N_{cs}, N_{cd} and N_{ov} where $N_{ov} = N_{it} + N_{cr}$.

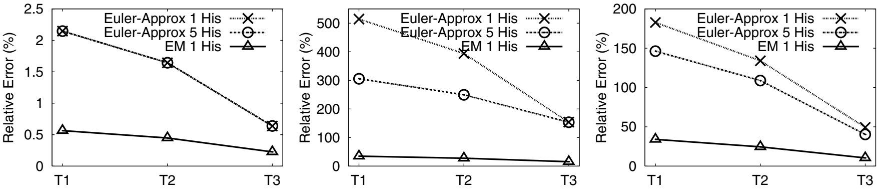
4.1 The Advantages of Min-Skew-Like Partition

To prove the advantages of Min-skew-like partition, in this part, we evaluate the performance of EM histogram in comparison with Euler-Approx and Multiscale Histogram.



(a) Nov (b) Ncs (c) Ncd

Fig. 5. Performance Comparison for Real Dataset: Ca_road



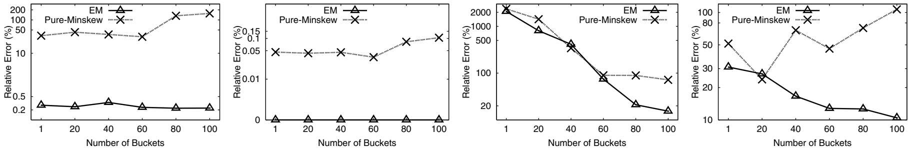
(a) Nov (b) Ncs (c) Ncd

Fig. 6. Performance Comparison for Synthetic Dataset: Zipf

First we compare the performance between EM histogram and Euler-Approx. Both algorithms can calculate the N_{ds} accurately. Figure 5 shows the experiment results for Ca_road dataset. Because the location distribution is not quite skew, the performance with different number of buckets of EM is quite similar. By Figure 5, we can see even using 1 bucket, EM histogram has a better result than that of Euler-Approx using 5 histograms in most cases.

Figure 6 shows the experiment results for synthetic dataset Zipf using different query set. 100 buckets are allocated to EM histogram. Again, even using 1 histogram, the EM still outperforms the Euler-Approx with 5 histograms. The performance difference between 2 algorithms is quite large compared with that of Ca_road dataset. This is because when the dataset follows non-uniform distribution, the assumptions made by Euler-Approx would be fail. On the other hand, EM histogram uses only a few buckets to capture the skew data distribution and local uniformity assumption is applied only in each bucket. Another problem of Euler-Approx shown in [8] is the accuracy of N_{ov} is fixed regardless of the number of histograms used (see Figure 5(a), 6(a)).

In fact, Multiscale histogram is a special case of EM histogram if $k = 1$ (the number of Euler histograms) and $B = 1$ (the number of buckets). On the other hand, we can also treat EM histogram as a solution for the last histogram in Multiscale histogram techniques. So for Ca_road dataset, the performance of EM (Figure 5) is also the performance of Multiscale histogram. Figure 7 for $B = 1$ is the experiment result of Multiscale histogram for dataset Zipf. We can see EM always outperforms Multiscale histogram.



(a) Nov (b) Nds (c) Ncs (d) Ncd

Fig. 7. Synthetic Data: Zipf

With Min-skew-like partition technique, EM histogram can capture the features of data objects accurately which are the critical information for the estimation accuracy.

4.2 The Advantages of Euler Histogram

To prove the advantages of Euler histogram, in this part, we evaluate the performance of EM histogram in comparison with Pure-Minskew method in which no Euler information is provided.

Figure 7 shows the experiment results of T_3 query set against the different number of buckets The results for T_1 and T_2 query set are similar to T_3 query set. With the increase of number of buckets, the N_{cs} and N_{cd} calculated by EM histogram are improved quickly. The N_{ov} with low relative error is not changed too much. We can always get the exact results for N_{ds} from EM histogram by using equation (3) and (5).

The accuracy of N_{cs} by Pure-Minskew method is increased with the increase of number of buckets. But it is interesting to note the performance of N_{cd} and N_{ov} are decreased. Another major concern about Pure-Minskew is, even the performance could be improved with the increase of number of buckets, a big number of buckets means more buckets have to be accessed in the on-line phase, which slows down the query performance.

With Euler histogram, EM histogram could use only a relative few number of buckets to accurately estimate the underlying data distribution. It is especially noteworthy for the on-line case. Pure-Minskew could not be a solution for selectivity estimation of middle-resolution relations.

4.3 Cost Evaluation

Storage Space. In theory, the storage space required is $(2n_1 - 1)(2n_2 - 1) + Bn_1n_2$. By our experiment, the space required for every 100 buckets is about 2.5KB (0.18% of one Euler histogram space) both for Ca_road dataset and Zipf dataset. So the storage space required for EM histogram is slight higher than those of the other 2 algorithms when using 1 Euler histogram. But from the experiments, we can see this is worthwile especially for non-uniform data. And more, 1 EM histogram even outperforms 5 Euler-Approx histogram in most cases but with much less space.

Histogram Construction Time. We evaluate the time to construct the EM histogram. The time costs for constructing 1 Euler histogram are similar to the Euler-Approx and Multiscale algorithms, about 40 to 41 seconds. For EM histogram, extra time is required to build buckets. With the resolution 360 × 180 spatial resolution, the time cost is about 43 seconds to build 100 buckets. The histogram construction time will be decreased with the decrease of spatial resolution.

Query Time. As analyzed in Section 3.4, the time for querying EM histogram is $O(B)$ (B is the number of buckets), which is irrelevant to the size of the Euler histogram and the underlying dataset. By our experiment, for every 10,000 window queries, it takes about 1 seconds for querying EM histogram with 100 buckets.

5 Conclusion and Remarks

In this paper, we investigate the problem of summarizing the spatial middle-resolution topological relations: contains, contained, overlap and disjoint. Spatial datasets could be various both in location distribution and scale distribution. A novel hybrid histogram technique, EM histogram, is presented as an accurate and effective tool to solve the problem. EM histogram use the concept of Min-skew partition in conjunction with Euler histogram approach. Our experiment results demonstrated that our approach may greatly improve the accuracy of existing techniques while retaining the costs efficiency.

As a possible future study, we will investigate the problem of non-aligned window query and explore other related research topics.

References

1. Greene, S., Tanin, E., Plaisant, C., Shneiderman, B., Olsen, L., Major, G., Johns, S.: The end of zero queries: Query preview for nasa's global change master directory. International Journal of Digital Libraries (1999) 70–90
2. Szalay, A., Kunzst, P., Thakar, A., Gray, J., Slutz, D., Brunner, R.: Designing and mining multi-terabyte astronomy archieves: The sloan digital sky survey. In: SIGMOD. (2000) 451–462
3. Sun, C., Agrawal, D., Abbadi, A.E.: Exploring spatial datasets with histograms. In: ICDE. (2002) 93–102
4. Beigel, R., Tanin, E.: The geometry of browsing. In: Proceedings of the Latin Ameriacn symposium on Theoretical Informatics,1998,Brazil. (1998) 331–340
5. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP operations in spatial data warehouses. In: SSTD. (2001) 443–459
6. Papadias, D., Tao, Y., Kalnis, P., Zhang, J.: Indexing spatial-temporal data warehouses. In: ICDE. (2002)
7. Poosala, V., Ioannidis, Y.E., Haas, P., Shekita, E.: Improved histograms for selectivity estimation of range predicates. In: SIGMOD. (1996) 294–305
8. Lin, X., Liu, Q., Yuan, Y., Zhou, X.: Multiscale histograms: Summarizing topological relations in large spatial datasets. In: VLDB. (2003) 814–825
9. Acharya, S., Poosala, V., Ranmaswamy, S.: Selectivity estimation in spatial databases. In: SIGMOD. (1999) 13–24

10. Aboulnaga, A., Naughton, J.F.: Accurate estimation of the cost of spatial selections. In: ICDE. (2000) 123–134
11. Jin, J., An, N., Sivasubramaniam, A.: Analyzing range queries on spatial data. In: ICDE. (2000) 525–534
12. Egenhofer, M.J., Herring, J.R.: Categorizing binary topological relations between regions, lines, and points in geographic databases. In Egenhofer, M.J., Mark, D.M., Herring, J.R., eds.: The 9-Intersection: Formalism and Its Use for Natural-Language Spatial Predicates. National Center for Geographic Information and Analysis, Report 94-1. (1994) 13–17
13. Grigni, M., Papadias, D., Papadimitriou, C.: Topological inference. In: IJCAI. (1995) 901–907
14. Ho, C.T., Agrawal, R., Megiddo, N., Srikant, R.: Range queries in olap data cubes. In: SIGMOD. (1997) 73–88
15. Chun, S.J., Chung, C.W., Lee, J.H., Lee, S.L.: Dynamic update cube for range-sum queries. In: VLDB. (2001) 814–825
16. TIGER: Tiger/line files. Technical report, U.S.Census Bureau, Washington, DC (2000)