Selectivity Estimation for Spatial Joins with Geometric Selections *†

Chengyu Sun Divyakant Agrawal Amr El Abbadi

Department of Computer Science
University of California, Santa Barbara
Email:{cysun, agrawal, amr}@cs.ucsb.edu

Abstract

Spatial join is an expensive operation that is commonly used in spatial database systems. In order to generate efficient query plans for the queries involving spatial join operations, it is crucial to obtain accurate selectivity estimates for these operations. In this paper we introduce a framework for estimating the selectivity of spatial joins constrained by geometric selections. The center piece of the framework is *Euler Histogram*, which decomposes the estimation process into estimations on vertices, edges and faces. Based on the characteristics of different datasets, different probabilistic models can be plugged into the framework to provide better estimation results. To demonstrate the effectiveness of this framework, we implement it by incorporating two existing probabilistic models, and compare the performance with the Geometric Histogram [AYS01] and the algorithm recently proposed by Mamoulis and Papadias [MP01].

Key words. spatial join, geometric selection, selectivity estimation

1 Introduction

Many applications such as geographical information systems (GIS) and computer-aided design (CAD) systems require a large amount of spatial objects to be stored, accessed and manipulated efficiently, which gives rise to the area of *spatial databases*. The data stored in spatial databases are *spatial objects* such as locations, road segments, and geographical regions, which can be abstracted as points, polylines and polygons in a 2D or 3D coordinate system.

One of the most important operations in spatial databases is *geometric selection* [RSV01], which in general can be classified as *windowing* and *clipping*. Given a set of spatial objects and a query rectangle,

*This work was partially supported by NSF grants EIA-9818320, IIS-98-17432, EIA-9986057, IIS-99-70700, and IIS-02-09112.

†An abridged version of this paper was published at EDBT2002.

a windowing query retrieves all the objects that intersect the query rectangle. A clipping query can be considered as a windowing query with a post-processing phase: once the objects that intersect the query rectangle are found, a clipping algorithm is applied to the objects to trim off parts of the objects that are outside the query rectangle.

Another important operation in spatial databases is *spatial join*, which retrieves *pairs* of objects from two datasets. An object pair is a valid result of a join if they satisfy certain criteria. Two commonly used criteria are *distance* and *intersection*, and the spatial joins based on these two criteria are referred to as distance join and intersection join. Distance join is typically used for finding pairs of objects that are within a given distance to each other. For example, the query "find all hotels that are within 10 miles to an airport" can be answered by the distance join of a hotel dataset and an airport dataset. For intersection joins, two objects constitutes a result pair if they intersect each other. Intersection join is commonly used in applications such as *map overlays*. For example, answering the query "find all French-speaking regions in Europe" involves the intersection join of a country map and a language coverage map. In this paper we only consider intersection joins.

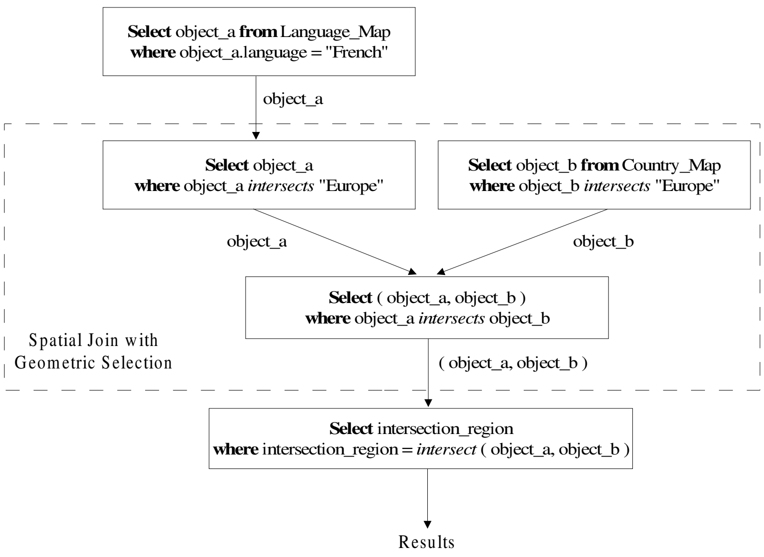


Figure 1: A Execution Plan for the Query "Find all French-speaking regions in Europe"

It can be observed that the query "find all French-speaking regions in Europe" is not a simple spatial join. In fact, it is a composite query involving several operations. Figure 1 shows a likely execution plan for this query in an SQL-like syntax, assuming the database holds two datasets, one of which is a language coverage dataset of the world and the other is a political/regional dataset of the world. Note that the first step on the dataset *Language_Map* is a simple attribute selection query which is well-studied in relational databases, and the last step applies a computation geometry algorithm which calculates the intersection regions of two sets of objects. What is more interesting from a spatial database perspective are the steps in the middle, which we term as a *spatial join with geometric selections* query, or a SJGS query for short. We are interested in these types of queries for two reasons. First of all, these types of queries are very common in spatial databases, and they are also very expensive in term of execution cost. For example, it is not uncommon in real world datasets that a single polygon consists of *thousands* of vertices. Therefore, optimizing SJGS queries is crucial for the performance of a spatial database management system. Secondly, to optimize queries that consist of SJGS subqueries, it is important to get accurate selectivity estimations of the SJGS subqueries. However, although much work has been done in estimating the selectivity of geometric selection *or* spatial join queries (discussed later in the related work), there is no effective way to combine these two methods. In this paper, we address this problem by developing a new approach that gives accurate selectivity estimation for SJGS queries. Our contributions are as follows:

- We demonstrate the use of *Euler Histogram* to solve a special case of SJGS queries. Euler histograms are powerful histograms that were previously used in selectivity estimation for spatial *intersection* [BT98] and *containment* [SAE01] queries.
- We generalize Euler Histogram to serve as a framework for selectivity estimation for common types of SJGS queries. The framework is capable of accommodating different probabilistic models for datasets with different characteristics.
- We incorporate two existing probabilistic models into our framework, and identify their strengths and weaknesses from both a theoretical and a practical perspective. An interesting finding of the analysis is that a mathematically correct model may not perform well in real world scenarios.

The rest of the paper is organized as follows: in Section 2 we formally define the problem, and Section 3 discusses the related work. In Section 4 we present a solution based on Euler Histogram for a special case of SJGS queries. In Section 5 we generalize Euler Histogram to serve as a selectivity estimation framework, and discuss two probabilistic models that can be used under this framework. An implementation of the generalized Euler Histogram is evaluated and compared with two existing algorithms in Section 6. Section 7 concludes the paper and discusses some future work.

2 Problem Definition

We formally define the problem of selectivity estimation for spatial join with geometric selection (SJGS) queries as follows:

Definition 2.1 *Let $A = \{a_1, ..., a_i, ...a_n\}$ and $B = \{b_1, ..., b_j, ..., b_m\}$ be two spatial datasets, where a_i and b_j are objects in A and B; and let S_A and S_B be two selection rectangles over A and B, respectively. The selectivity estimation problem for the join of A and B with selections S_A and S_B is to estimate the number of object pairs (a_i, b_j) that satisfy the following three conditions:*

1. *a_i intersects S_A, and*
2. *b_j intersects S_B, and*

3. a_i *intersects* b_j*.*

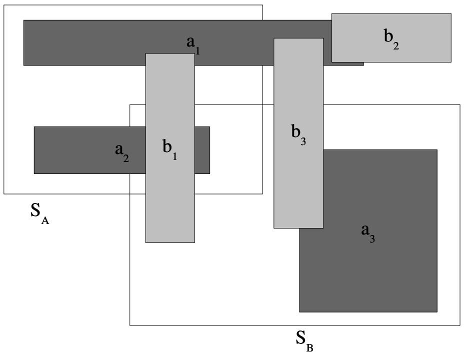


Figure 2: An Example of a SJGS Query

Figure 2 shows two datasets A and B, and two selection windows S_A and S_B of a SJGS query. From Definition 2.1, we can see that the object pairs (a_1, b_1),(a_1, b_3) and (a_2, b_1) are all valid results of the query. On the other hand, (a_1, b_2) is not a valid result because b_2 does not intersect S_B, and similarly, (a_3, b_3) is not a valid result because a_3 does not intersect S_A.

There are several points we would like to clarify about Definition 2.1 and our system assumptions in general.

First of all, the spatial objects considered in this definition and the rest of the paper are rectangles, including the special cases where the rectangles are reduced to vertical/horizontal line segments or points. For simplicity and readability, we also assume the objects are in a 2-dimensional space. In Section 4, we will show that our technique can be easily extended to higher dimensions. However, we do acknowledge that since our technique is based on uniform partitioning, it is subject to the *dimensionality curse*, namely, the number of partitions increases exponentially as the dimensionality increases.

Secondly, a *data space* for a dataset is the bounding rectangle that encloses all the objects in the dataset, and we assume that both datasets A and B have a common data space. For example, a rainfall map of Orange County and a land cover map of Orange County will have the common data space *Orange County*, or the rectangle $(-118.27W, -117.34E, 32.58S, 34.02N)$ in a longitude-latitude coordinate system. Note that this assumption is not restrictive, since if two datasets have different data spaces, we can always take the bounding rectangle of the two data spaces as the common data space.

Finally, in this paper we concentrate on the selectivity estimation problem where the two selection windows select the same region, or $S_A = S_B = S$. The solution developed in this paper can be applied to two common classes of SJGS queries:

- *Spatial Joins with Same Selections.* This class of query is the most common in applications such as map overlays. The sample query in Section 1, "finding all French-speaking regions in Europe", is an example of this class of query. Full dataset joins discussed in previous work [AYS01] can also be considered as a special case of this class of query where the selection windows are the data space.

- *Spatial Joins with Clipping Selections.* Even for SJGS queries with different selection windows, our technique is still applicable if the selections are *clipping* instead of *windowing*. For example, Figure 2 shows that (a_1, b_1) *and* (a_1, b_3) are valid results under windowing selections. However, note that for clipping selections, these object pairs are *not* valid results because after clipping, a_1 no longer intersects b_1 or b_3. In fact, for selectivity estimation, a spatial join with different clipping selections S_A and S_B can be considered the same as a spatial join with the same selection window S, where S is the intersecting region of S_A and S_B, or $S = S_A \cap S_B$.

3 Related Work

Selectivity Estimation for Spatial Selection Queries. Much work has been done in this area. Among the many existing techniques, the histogram-based approaches that target region objects [APR99, AN00, JAS00] are more closely related to our work. In [APR99], Acharya et. al. proposed the *MinSkew* algorithm. The MinSkew algorithm starts with a density histogram of the dataset, which effectively transforms region objects to point data. The density histogram is further split into more buckets until the given bucket count is reached or the sum of the variance in each bucket cannot be reduced by additional splitting. In [AN00], the SQ Histogram is proposed with an emphasis on polygon datasets. The SQ algorithm first builds a hierarchy of buckets based on a Quad-tree, and the objects are assigned to the buckets according to their center positions and areas. Similar buckets are then merged to meet the bucket count requirement. The CD Histogram is proposed in [JAS00]. Typically when building a histogram for region objects, an object may be counted multiple times if it spans across several buckets. The CD algorithm addresses this problem by keeping four sub-histograms which correspond to the four corner points of a rectangle. A bucket in a sub-histogram stores the number of corresponding corner points that fall in the bucket, so even if a rectangle spans several buckets, it is counted exactly once in each sub-histogram.

Euler Histogram for Spatial Dataset Browsing. Browsing is closely related to selectivity estimation, where a user is interested in visualizing the data distribution of a dataset without accessing the actual data. Euler Histograms were first introduced by Beigel and Tanin in [BT98] for browsing all objects in a spatial dataset that intersect a selected region. [SAE01] extends the work to spatial relations including *contains*, *contained* and *overlap*. The mathematical foundation of the Euler Histogram is based on Euler's Formula in graph theory, hence the name *Euler* Histogram. As in the CD Histogram, Euler Histograms also addresses the multiple-count problem, but does not require the objects to be rectangular.

Selectivity Estimation for Spatial Joins. Compared to selectivity estimation for spatial selections, this area has only recently attracted much research attention. In [FSTJ00], Faloutsos et. al. proposed a parametric approach to estimate distance join selectivity based on the observation that the data distribution of many real world datasets follow a *power law*. By pre-computing a set of parameters of the data distribution, the selectivity of a distance join can be estimated in constant time. For intersection joins, the state of the art are two techniques recently proposed in [AYS01] and [MP01], both of which are histogram-based approaches. The technique proposed in [AYS01], called Geometric Histogram (GH), is based on the observation that the intersection of two rectangles always results in four *intersection points* (the four corner points of the intersection region), therefore the selectivity of a spatial join can be estimated by first estimating the number of intersection points then dividing the result by 4. Another technique is proposed in by Mamoulis and Papadias [MP01], which we will refer to as the MP algorithm. To the best of our knowledge, MP algorithm is the first technique that addresses the problem of spatial joins with geometric selections. A sophisticated mechanism is developed in [MP01] to calculate the *work space* of two or more selection windows, in which objects from different datasets may intersect. However, the basic assumption of the MP algorithm is that the objects intersecting a grid cell are not only uniformly distributed in the space domain, but also have similar widths and heights. We believe this assumption is too strong for many real world datasets.

4 Basic Euler Histogram

In this section we give a brief introduction to the Euler Histogram, and show how it can be used to solve a special case of SJGS query selectivity estimation where the datasets to be queried satisfy certain restrictions.

4.1 Euler Histogram

The conventional approach to build a histogram for region objects is as follows: given a grid of the data space, allocate a set of buckets with each bucket corresponding to a grid cell; and for each object, if it intersects a bucket, then increase the value stored in the bucket by 1. Conventional histograms suffer from a severe drawback which we call the *multiple-count problem.* Unlike point objects which either fall inside a histogram bucket or not, a region object may span several buckets and therefore is counted multiple times. The multiple-count problem is illustrated in Figure 3(a) and (b), where Figure 3(a) shows a dataset with a large object spanning four buckets, and Figure 3(b) shows the conventional histogram for this dataset. Note that for a spatial selection query covering the four buckets, the single object will be counted four times, resulting in a 300% over estimation.

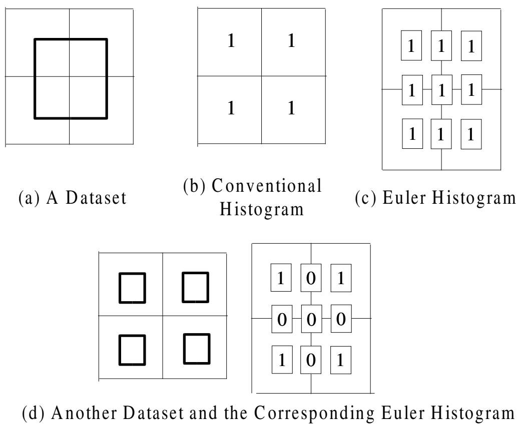


Figure 3: Comparison between Conventional Histogram and Euler Histogram

Euler Histograms are designed to address the multiple-count problem. Compared to the conventional histogram, an Euler histogram allocate buckets not only for grid cells, but also for grid cell edges and vertices. For example, the Euler histogram for the dataset in Figure 3(a) is shown in Figure 3(c), and for comparison, another dataset with four small objects and its corresponding Euler histogram is shown in Figure 3(d). Note that with additional buckets for grid cell edges and vertices, it is now possible to distinguish between a large object spanning several cells and several small objects contained in each cell.

In [BT98], Beigel and Tanin presented an algorithm to estimate the selectivity of geometric selection queries using Euler histograms. The mathematical foundation of the algorithm is based on Euler's Formula in graph theory, therefore we refer to this type of histograms as the *Euler Histograms.* The development of the algorithm is lengthy and was discussed in full detail in [BT98], so here we only give the algorithm and highlight some important results:

Given a selection window S, the selectivity of S can be calculated with an Euler histogram as follows:

$$Selectivity(S) = \sum_{0 \leq k \leq d} (-1)^k F_k(S) \tag{1}$$

where $F_k(S)$ is a *k-dimensional face* inside S. In particular, a 0-dimensional face is a vertex, a 1-dimensional face is an edge and a 2-dimensional face is a cell.

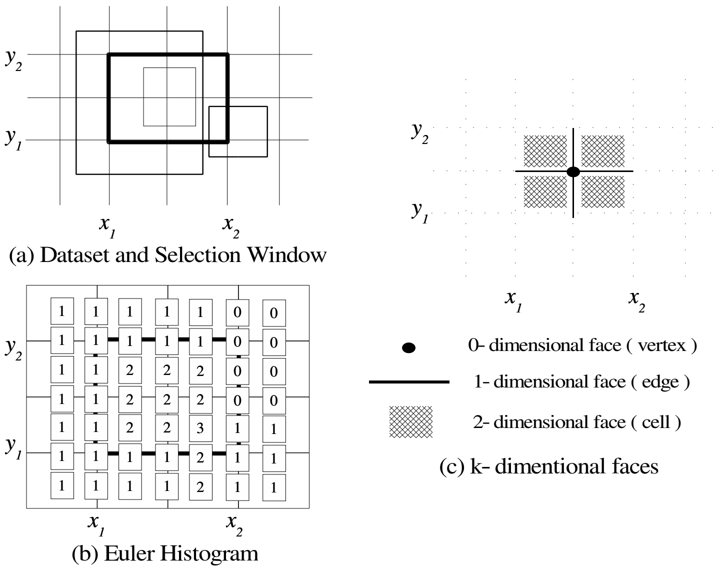


Figure 4: An Example of Beigel-Tannin's Algorithm

An example of Beigel-Tannin's algorithm is illustrated in Figure 4. Figure 4(a) shows a dataset with three rectangular objects, and a selection window S at (x_1, x_2, y_1, y_2). Figure 4(b) shows the Euler histogram for this dataset. The one 0-d face, four 1-d faces and four 2-d faces that are *inside* (excluding boundary) Q are highlighted in Figure 4(c). So from Equation 1, the selectivity of S is $(-1)^0 \times 2 + (-1)^1 \times (2+2+2+2) + (-1)^2 \times (2+2+2+3) = 3$.

In [BT98], Beigel and Tanin established two important properties of Equation 1:

1. If the selection window aligns with the grid, the estimation given by Equation 1 is *without any errors.*

2. Equation 1 holds for any d-dimensional spaces where $d \geq 1$.

4.2 A Special Case of the SJGS Estimation Problem

In Section 2 we defined the problem of selectivity estimation for SJGS queries, and now we consider a special case of this problem. Namely, given two datasets A and B, and a selection windows S, we require that all objects be "quantized" to align with a given grid. We call this assumption the *Quantization Assumption*, which is illustrated in Figure 5. Notice that objects $(0.2, 1.4, 2.5, 3.5)$ and $(1.6, 2.6, 1.5, 2.7)$ become $(0, 2, 2, 4)$ and $(1, 3, 1, 3)$ after quantization.

The Quantization Assumption is clearly a very strong assumption. Under this assumption, not only the objects are quantized, but selection windows are implicitly quantized, too. Using the two quantized objects in Figure 5(b) as an example, we can see that a selection window at $(0.5, 1.5, 1.5, 2.5)$ (the dotted rectangle in Figure 5(b)) will have the same results as a query window at $(0, 2, 1, 3)$.

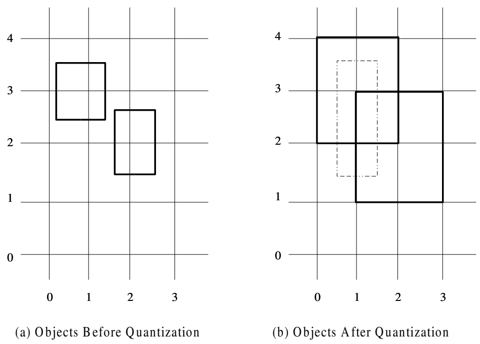


Figure 5: Two Objects Before and After Quantization

Under the Quantization Assumption, we can now use the Euler Histogram to estimate the selectivity of a SJGS query as follows:

Given two datasets A, B and a grid of the data space, we construct an Euler histogram for each dataset and call them H_A and H_B. Let $h^A_{i,j,k}$ and $h^B_{i,j,k}$ be buckets in H_A and H_B, respectively. The meanings of the subscripts of $h_{i,j,k}$ are the following: i is the row position of h in the histogram, j is the column position, and k means that h corresponds to a k-dimensional face of the grid. For a SJGS query Q with a selection window S, the selectivity of the query can be calculated as

$$Selectivity(Q) = \sum_{i,j,k} (-1)^k [h^A_{i,j,k}(S) \times h^B_{i,j,k}(S)] \qquad (2)$$

where $h^A_{i,j,k}$ and $h^B_{i,j,k}$ are buckets of H_A and H_B, respectively, that are *inside* the selection window S. We further claim that when the Quantization assumption holds, the estimation given by Equation 2 is *without any errors.*

Before we prove the correctness of the algorithm, let's first look at an example. Figure 6(a) shows two datasets A and B. A consists of three objects, and B consists of two objects. The selection window is at (x_1, x_2, y_1, y_2). The Euler histograms for A and B are shown in Figure 6(b). For better readability we removed all buckets except those that are *inside* the selection window. From Equation 2, the selectivity of this SJGS query is

$$\begin{aligned} & (-1)^0(1\times 1)+(-1)^1(2\times 1+1\times 1+1\times 1+1\times 2)+(-1)^2(2\times 1+2\times 1+1\times 2+2\times 2) \\ = \; & 1-6+10 \\ = \; & 5 \end{aligned}$$

Note that the larger object in B intersect all three objects in A, and the smaller object in B intersects two objects in A, so the selectivity of the query is indeed 5.

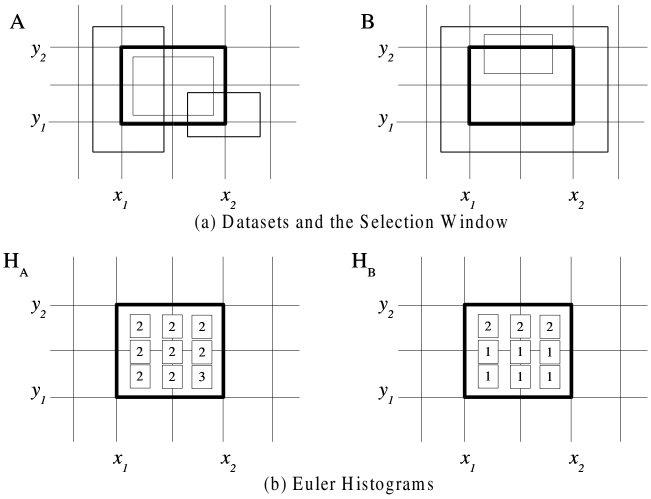


Figure 6: Selectivity Estimation for the Special Case

We now prove the correctness of the algorithm and justify our claim that under the Quantization Assumption, the estimation is *without any errors*. First we make the following observation: if two objects both intersect the selection window as well as each other, then their intersection region also intersects the selection window. This observation is formalized as the following lemma:

Lemma 4.1 *If two rectangular objects a and b both intersect rectangle S as well as each other, then their intersection region, denoted as ab, also intersects S.*[1]

Proof. Let a be at $(x_{a1}, x_{a2}, y_{a1}, y_{a2})$, b be at $(x_{b1}, x_{b2}, y_{b1}, y_{b2})$ and S be at $(x_{S1}, x_{S2}, y_{S1}, y_{S2})$, as shown in Figure 7. If a intersects b, then the coordinates of their intersecting region, ab (the shaded area in Figure 7), are $(max(x_{a1}, x_{b1}), min(x_{a2}, x_{b2}), max(y_{a1}, y_{b1}), min(y_{a2}, y_{b2}))$.

Now assume ab and S do not intersect, then there are four possible cases: ab is to the left of S, ab is to the right of S, ab is above S, or ab is below S. If ab is to the left of S, then $min(x_{a2}, x_{b2}) < x_{S1}$. So either

[1]Lemma 4.1 is such a straight-forward observation that it must have been proved before. However since we could not track down a reference, we proved it anyway.

$x_{a2} < X_{S1}$, meaning a is to the left of S; or $x_{b2} < x_{S1}$, meaning b is to the left of S. Both cases contradict the fact that a and b both intersect S. Similarly, we can show that ab cannot be to the right of, above or below S. Therefore by contradiction we prove that ab must intersect S. □

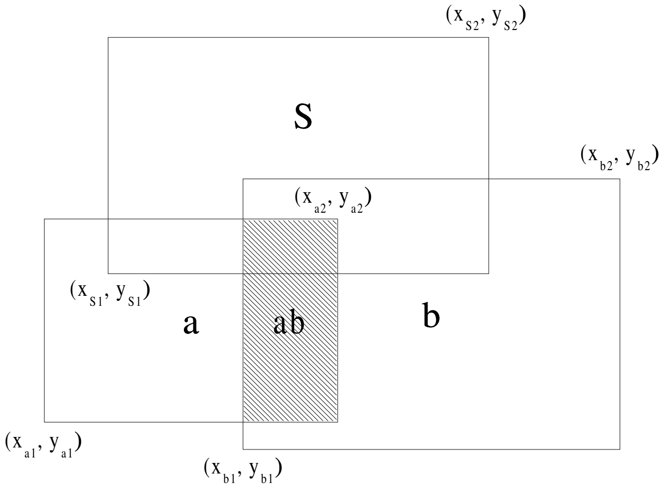


Figure 7: Lemma 4.1

Based on Lemma 4.1, the correctness of the algorithm can be proved as follows:

Theorem 4.1 *Let Q be an SJGS query with selection window S over datasets A and B. Under the Quantization Assumption, the selectivity estimate of Q produced by Equation 2 is without any errors.*

Proof.

Step 1. We first construct a new dataset AB which consists of all the intersecting regions between the objects from A and the objects from B. Formally, $AB = \{\ ab \mid ab = a \cap b, a \in A, b \in B\}$.

Step 2. We construct an Euler histogram H_{AB} for the new dataset AB. Let $h^{AB}_{i,j,k}$ be a bucket in H_{AB}, and $h^{A}_{i,j,k}$ and $h^{B}_{i,j,k}$ be the corresponding buckets in H_A and H_B, respectively. If $h^{A}_{i,j,k} = x$ and $h^{B}_{i,j,k} = y$, then by the Quantization Assumption (if two objects intersect the same face they must intersect each other) there are $x \times y$ intersecting regions; and by Lemma 4.1, these $x \times y$ intersecting regions intersect $h^{AB}_{i,j,k}$ (0-d and 1-d faces can be considered as special cases of S in Lemma 4.1, where $x_{S1} = x_{S2}$ or $y_{S1} = y_{S2}$ or both), so $h^{AB}_{i,j,k} = x \times y = h^{A}_{i,j,k} \times h^{B}_{i,j,k}$.

Step 3. If (a, b) is a valid result of Q with selection window S, then from Lemma 4.1, ab intersects S; and the reverse is also true (if ab intersects S then (a, b) is clearly a valid result of Q). This means that the selectivity of the SJGS query Q over A and B is the same as the selectivity of the geometric selection query S over AB, or $Selectivity(Q_{A,B}) = Selectivity(S_{AB})$. From Equation 1, we have $Selectivity(S_{AB}) = \sum_{i,j,k}(-1)^k h^{AB}_{i,j,k}$, and it is proved in [BT98] that when the selection window aligns with the grid (Quantization Assumption), the selectivity estimation is without any errors. Note that from Step 2 we have $h^{AB}_{i,j,k} = x \times y = h^{A}_{i,j,k} \times h^{B}_{i,j,k}$, so $Selectivity(Q_{A,B}) = Selectivity(S_{AB}) = \sum_{i,j,k}(-1)^k h^{AB}_{i,j,k} =$

$\sum_{i,j,k}(-1)^k[h^A_{i,j,k} \times h^B_{i,j,k}]$, which is exactly Equation 2. Therefore the selectivity estimate of Q produced by Equation 2 is *without any errors.* □

Based on Theorem 4.1 and the discussion in Section 2, we can also conclude that under the quantization assumption, Euler Histogram provides *exact* results for SJGS queries with clipping selections, even when the selection windows for the two datasets are different.

5 Generalized Euler Histogram

In this section we generalize the Euler Histogram approach to handle objects and selection windows that do not align with a given grid. The generalized Euler Histogram serves as a framework for estimating the selectivity of SJGS queries. We discuss two existing probabilistic models, and show how they can be incorporated into the framework to form a complete solution.

5.1 Euler Histogram Framework

We first revisit Equation 2. In a 2-dimensional space, Equation 2 can be written as follows:

$$Selectivity(Q) = \sum_{i,j}[h^A_{i,j,2}(S) \times h^B_{i,j,2}(S)] - \sum_{i,j}[h^A_{i,j,1}(S) \times h^B_{i,j,1}(S)] + \sum_{i,j}[h^A_{i,j,0}(S) \times h^B_{i,j,0}(S)] \quad (3)$$

where $h^A_{i,j,2}(S) \times h^B_{i,j,2}(S)$, $h^A_{i,j,1}(S) \times h^B_{i,j,1}(S)$ and $h^A_{i,j,0} \times h^B_{i,j,0}(S)$ are the estimated number of intersection regions that intersect a 2-d face (cell), a 1-d face (edge) or a 0-d face (vertex), respectively.

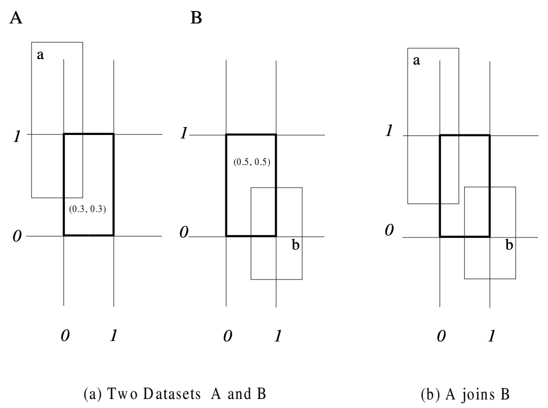


Figure 8: Comparison between Euler Histograms with and without Fractions

Under the Quantization Assumption, when two objects intersect the same face, they must intersect each other. So if there are h^A objects from dataset A that intersect a face (i, j), and there are h^B objects from dataset B that intersect the same face, then the number of intersection regions that intersect face (i, j) is *exactly* $h^A \times h^B$. However, with objects that do not align with the histogram grid, the number of intersection regions may be less than $h^A \times h^B$. For example, Figure 8 shows two objects a and b, both of them intersect the 2-dimensional face at $(0,1,0,1)$, but do not intersect each other. To address this problem, we generalize the Euler Histogram approach by introducing an *intersection factor* p, so in a 2-dimensional space, Equation 2 becomes

$$Selectivity(Q) = \sum_{i,j,k=0,1,2} (-1)^k [h^A_{i,j,k}(S) \times h^B_{i,j,k}(S) \times p_{i,j,k}] \tag{4}$$

Intuitively, $p_{i,j,k}$ is the *probability* that one set of objects intersect another set of objects inside face (i,j,k). For 0-d faces, or vertices, $p_{i,j,0}$ is always 1; for higher dimensional faces, the value of $p_{i,j,k}$ is between 0 and 1, depending on the sizes and spatial distributions of the objects inside the face.

Note that Equation 4 does not specify how $p_{i,j,k}$ should be computed. Instead, Equation 4 serves as a framework where different probabilistic models can be applied. For certain datasets, some probabilistic models may capture the data distribution better than the others, which will result in more accurate selectivity estimation. In the next subsection, we will discuss two existing probabilistic models. Our discussion will be focus on the case of 2-d faces, which can be easily generalized to handle 1-d faces.

5.2 Probabilistic Models

Given two sets of objects $A = \{a_1, ..., a_i, ..., a_{n_1}\}$ and $B = \{b_1, ..., b_j, ..., b_{n_2}\}$ that are enclosed in a 1×1 space, our goal is to find the number of object pairs (a_i, b_j) where a_i intersect b_j. Clearly, finding the exact number of intersecting pairs requires keeping track of the shape and the position of each object, which is not efficient for selectivity estimation purposes. So the common approach is to keep a set of parameters which summarize the information about each dataset, and rely on a probabilistic model to provide an estimated result.

In [MP01], Mamoulis and Papadias proposed keeping three parameter for each dataset: the number of objects n, the average object height $\overline{h}$ and the average object width $\overline{w}$. The, the number of intersecting pairs N is estimated using the following equation:

$$N = n_1 \times n_2 \times min(1, (\overline{h_1} + \overline{h_2})) \times min(1, (\overline{w_1} + \overline{w_2})) \tag{5}$$

In [AYS01], An et al. proposed another probabilistic model for the Geometric Histogram (GH) algorithm for spatial joins without selections. The GH model keeps four parameters: the total number of corner points C, the sum of the areas of the objects O, the sum of the lengths of the vertical edges V, and the sum of the lengths of the horizontal edges H. Under the GH model, the number of intersecting pairs can be estimated as follows:

$$N = \frac{C_1 \times O_2 + C_2 \times O_1 + V_1 \times H_2 + V_2 \times H_1}{4} \tag{6}$$

For comparison purpose, we rewrite Equation 6 as follows:

$$N = n_1 \times n_2 \times (\overline{a_1} + \overline{a_2} + \overline{h_1} \times \overline{w_2} + \overline{h_2} \times \overline{w_1}) \tag{7}$$

where $\overline{a}$ is the average area of the objects. Note that

$$C = 4 \times n$$

$$O = \overline{a} \times n$$
$$V = 2 \times \overline{h} \times n$$
$$H = 2 \times \overline{w} \times n$$

So Equation 6 and 7 are equivalent.

In order to decide which model is more suitable for the generalized Euler Histogram, we compare the performance of the GH model and Mamoulis-Papadias's model (MP model) under two conditions:

- $\overline{h_1} + \overline{h_2} >= 1$ and $\overline{w_1} + \overline{w_2} >= 1$

  In this case, Equation 5 reduces to $N = n_1 \times n_2$. Intuitively, the result of the MP model may be an over estimate, since it is possible that some objects from one dataset do not intersect *all* objects in the other dataset. On the other hand, the GH model is likely to perform even worse. For example, assume dataset A has one object $(0.1, 0.9, 0.1, 0.9)$ and dataset B also has one object $(0.1, 0.9, 0.1, 0.9)$. According to Equation 7, the estimated number of intersections is

  $$0.8 \times 0.8 + 0.8 \times 0.8 + 0.8 \times 0.8 + 0.8 \times 0.8 = 2.56$$

  while the correct answer is 1.

  The poor performance of the GH model in this case is because the GH model estimates four types of intersection points *independently*. When $\overline{h_1} + \overline{h_2} >= 1$ and $\overline{w_1} + \overline{w_2} >= 1$, the correlations among different types of intersection points cannot be ignored. For example, if one object falls completed inside another object, then no matter what their edge lengths are, the probability of edge intersection should be 0.

- $\overline{h_1} + \overline{h_2} < 1$ and $\overline{w_1} + \overline{w_2} < 1$.

  In this case, Equation 5 is reduced to

  $$N = n_1 \times n_2 \times (\overline{h_1} \times \overline{w1} + \overline{h_2} \times \overline{w_2} + \overline{h_1} \times \overline{w_2} + \overline{h_2} \times \overline{w_1}) \tag{8}$$

  Comparing Equation 8 to Equation 7, we can observe that the only difference is the MP model uses $\overline{h_1} \times \overline{w1}$ and $\overline{h_2} \times \overline{w2}$ while the GH model uses $\overline{a_1}$ and $\overline{a_2}$.

  Note that $(\overline{h_1} \times \overline{w1} + \overline{h_2} \times \overline{w2})$ and $(\overline{a_1} + \overline{a_2})$ are used to compute the probability of corner points inside objects, so we expect the GH model to perform better than the MP model. We can further predict that the MP model is likely to over estimate since $\overline{h} \times \overline{w} \geq \overline{a}$ holds for all datasets.

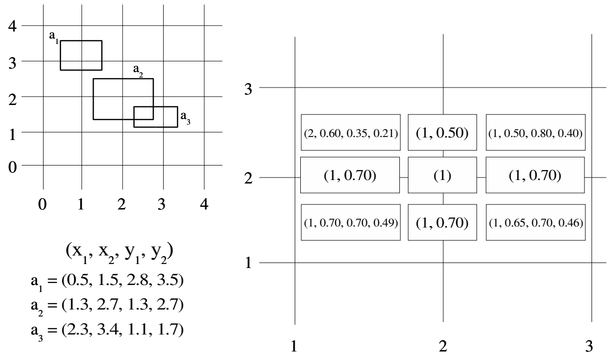
5.3 Implementation of Generalized Euler Histogram

We now present an implementation of the generalized Euler Histogram based on the discussion above. In particular, we adaptively use the GH model and the MP model depending on the properties of the objects in a histogram bucket.

Given a dataset and a grid of the data space, a generalized Euler histogram can be constructed as follows:

- allocate a bucket for each vertex, edge and cell of the grid.
- at a vertex, store the tuple $\langle n \rangle$, where n is the number of objects that intersect the vertex.
- at an edge, store the tuple $\langle n, \overline{e} \rangle$, where n is the number of objects intersecting the edge, and $\overline{e}$ is the average projection of the objects on this edge.
- at a cell, store the tuple $\langle n, \overline{h}, \overline{w}, \overline{a} \rangle$, where n is the number of objects intersecting the cell, and $\overline{h}$, $\overline{w}$ and $\overline{a}$ are the average height, width and area of the intersection regions between the objects and the cell.

An example of a generalized Euler histogram is shown in Figure 9.



(a) DataSet (b) Generalized Euler Histogram

Figure 9: A Generalized Euler Histogram

To estimate the selectivity of a given query, we use Equation 4 combined with a hybrid probabilistic model of GH and MP. For edges, the intersection factor $p_{i,j,1}$ is

$$p_{i,j,1} = min(1, \overline{e_1} + \overline{e_2}) \text{ (MP Model)}$$

and for cells, the intersection factor $p_{i,j,2}$ is

$$p_{i,j,2} = \begin{cases} n1 \times n2 & \text{if } \overline{h_1} + \overline{h_2} \geq 1 \text{ and } \overline{w_1} + \overline{w_2} \geq 1 \text{ (MP model)} \\ \overline{a_1} + \overline{a_2} + \overline{h_1} \times \overline{w_2} + \overline{h_2} \times \overline{w_1} & \text{otherwise (GH Model)} \end{cases}$$

6 Performance Evaluation

To evaluate the effectiveness of the generalized Euler Histogram (EH), we compare it with the Geometric Histogram (GH) algorithm [AYS01] and Mamoulis-Papadias's (MP) algorithm [MP01]. In this section, we describe our experimental setup and report the performance results.

6.1 Experimental Setup

6.1.1 Datasets

Our experiments were conducted on a PIII 800 PC with the following dataset:

- Land Cover Data for Wyoming (**LANDC**) [Ana96a]. This dataset contains land cover (mostly vegetation types) information for the state of Wyoming at 1:100000 scale.
- Land Ownership and Management for Wyoming (**LANDO**) [Ana96b]. This dataset contains land cover ownership and management information for the state of Wyoming at 1:100,000 scale.
- Digital Soils Map of Wyoming (**SOIL**) [MA99]. This dataset represents soils of Wyoming at 1:100,000-scale. The original data source consists of 23 datasets, each for a county of Wyoming. We combined these small datasets into a single dataset.
- Sequoia 2000 Benchmark, Polygon Dataset (Level 1) (**SQ1**) [SFGM93]. This dataset is one of the polygon datasets from the Sequoia 2000 Benchmark, which consists of urban and built-up land features.
- Sequoia 2000 Benchmark, Polygon Dataset (Level 2-9) (**SQX**) [SFGM93]. This dataset is the combination of several Sequoia 2000 polygon datasets, and consists of agricultural, water and other land features.
- Synthetic Datasets. Most objects in the real world datasets are small with respect to the size of the data space. To better understand the performance of each algorithm under different situations, we also include two synthetic datasets **SZSKEW1** and **SZSKEW2**, both of which contain significant numbers of large objects. The objects in SZSKEW1 and SZSKEW2 are square objects, with the centers the objects uniformly distributed in a 360×180 space. The side lengths of the objects follows a $Zipf$ distribution between 1 and 180.

Some statistics of the datasets are listed in Table 1. Note that all the polygon objects are represented by their minimal bounding rectangles (MBRs).

6.1.2 Queries and Performance Metrics

In each experiment we run a set of SJGS queries with randomly generated selection windows. The aspect ratios of the selection windows are uniformly distributed between 0.3 and 3. The queries are further divided into groups, with 20 queries in each group. The queries in the same group have the same *area selectivity*, which is the area of the selection window normalized to the area of the data space. The area selectivities used in the experiments are 25%, 10%, 1%, 0.1%, and 0.01%.

For each query group, the estimation accuracy of the algorithm is measure by the *Average Relative Error*, which is defined as follows: If e_i is the estimated answer and r_i is the actual answer for a given query q_i, the average relative error for a query group Q is $(\sum_{q_i \in Q} |r_i - e_i|)/(\sum_{q_i \in Q} r_i)$.

Dataset	Number of Objects	Data Space		Object	
		Height	Width	Avg. Height	Avg. Width
LANDC	14,731	4,115,124	7,104,683	54,269	74,970
LANDO	33,860	4,115,120	7,104,973	14,812	20,877
SOIL	29,662	4,006,340	6,997,317	19,925	28,142
SQ1	22,094	1,301,265	859,734	904	885
SQX	36,492	1,363,840	871,310	2,257	2098
SZSKEW1	100,000	179.999	359.990	8.901	8.901
SZSKEW2	100,000	179.997	359.993	8.944	8.944

Table 1: Datasets

6.1.3 Implementation Details

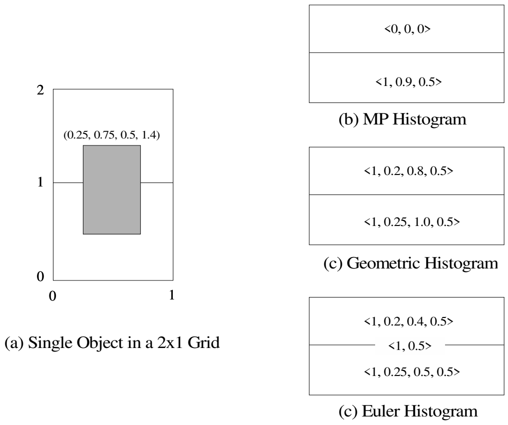


Figure 10: Three Types of Histograms

Histogram Construction. A simple example is shown in Figure 10 to illustrate the three types of histograms MP, GH and EH. Figure 10(a) shows a single object inside a 2×1 grid. The coordinates of the object is $(0.25, 0.75, 0.5, 1.4)$, and the MP, GH and EH histogram for this object is shown in Figure 10(b), (c) and (d). Besides the different probabilistic models discussed in Section 5.2 and 5.3, we also note from Figure 10 another important difference between MP histograms and GH/EH histograms: for MP histograms, an object spanning several buckets will only be counted in the bucket where the center of the object lies; for GH and EH histograms, the object will be split into *pieces*, and the pieces of the objects are counted separately.

Selectivity Estimation. The original Geometric Histogram was designed for full datasets joins. In our implementation, we made a slight modification so it can handle SJGS queries. Namely, the construction of a geometric histogram is still the same as described in [AYS01], but during estimation, we only estimate the number of intersection points inside the selection window. This modification may incur some additional estimation error, because some intersection points may be outside the selection window. The probability of an intersection point falling outside the selection window is given by the following equation:

$$1 - \frac{H \times W}{(H + \overline{h}) \times (W + \overline{w})} \tag{9}$$

where H and W are the height and width of the selection windows, and $\overline{h}$ and $\overline{w}$ are the average height and width of the objects that intersect the selection window. In most test cases, this error is negligible because $H >> \overline{h}$ and $W >> \overline{w}$.

Experiment Conventions. In our experiments, we follow the convention in [AYS01] and use a parameter *Level*, denoted as l, to represent the granularity of the grid. Formally, a Level l grid is an $n \times n$ grid of the data space where $n = 2^l$. For example, a level 0 grid is a 1×1 grid and a level 2 grid is a 4×4 grid.

6.2 Experimental Results

In this section we study the performance of MP, GH and EH with respect to the grid granularity and the sizes of the selection windows.

6.2.1 Impact of Grid Granularity

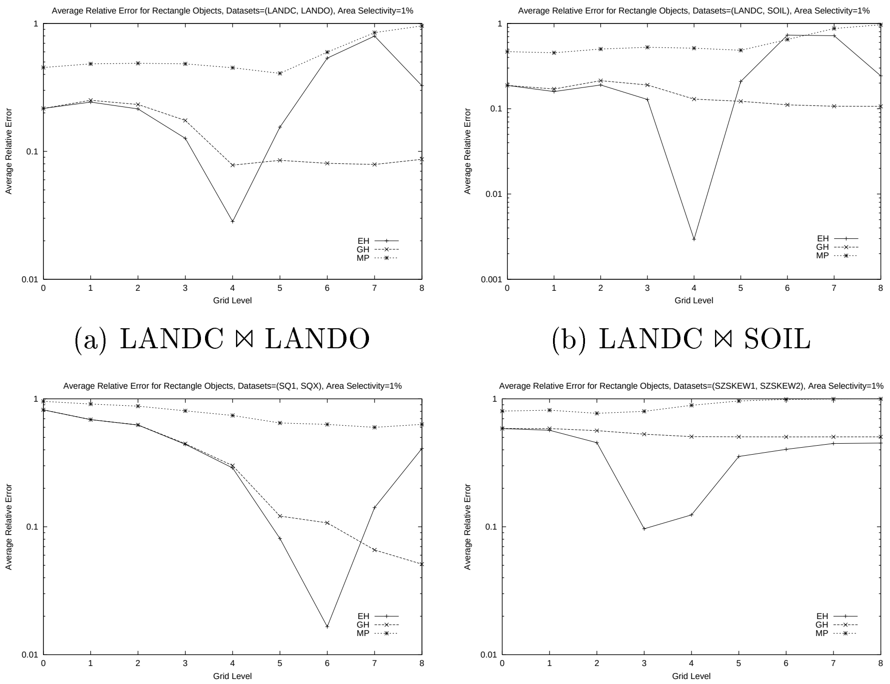
The estimation accuracy as a function of the grid level is shown in Figure 11. The Y-coordinate of a data point is the average relative estimation error computed from 20 randomly generated selection windows with the same area selectivity 1%. The X-coordinate of a data point is the grid level of the histogram.

From Figure 11, we make the following observations:

Geometric Histogram (GH) is the most stable algorithm among the three algorithms tested. In most cases, the estimation errors of GH consistently decreases as the grid granularity increases. As we discussed in Section 5.2, the probabilistic model of GH relies on the assumption that the heights and widths of the objects are relatively small with respect to the height and width of the selection window. So for datasets with small objects such as SQ1 and SQX, GH performs very well (Figure 11(c)), but for datasets with large objects such as SZSKEW1 and SZSKEW2, GH performs rather poorly (Figure 11(d)).

Mamoulis-Papadias's algorithm (MP) has high errors rates at all grid levels. There are two reasons for the poor performance of MP. First of all, the probabilistic model of MP is less accurate than the GH model except for extreme cases where the sizes of the objects are comparable to the size of the selection window. Secondly, unlike GH or EH, MP does not split objects which span several buckets, therefore benefits less from finer grid granularity. Furthermore, with finer grid granularity, although the centers of the objects are more uniformly distributed, the variance of the heights and the widths of the objects may actually increase, which also has a negative impact on the estimation accuracy of MP.

In all four test cases, Euler Histogram (EH) achieves the lowest estimation error, ranging from 0.3% in Figure 11(b) to about 10% in Figure 11(d). At the lower grid levels, EH uses the same probabilistic model as GH, but in addition to keeping information on grid cells, EH also keeps information on grid edges and



(a) LANDC ⋈ LANDO (b) LANDC ⋈ SOIL

(c) SQ1 ⋈ SQX (d) SZSKEW1 ⋈ SZSKEW2

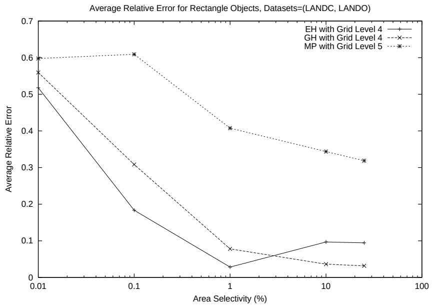
Figure 11: Average Relative Error for Queries with 1% Area Selectivity

vertices. Another important difference between EH and GH is that EH uses the *interiors* of the intersection regions, while GH uses the *corner points* of the intersection regions. As Figure 7 shows, the interior of an intersection region always intersects the selection windows, but some of the corner points may be outside the selection window. The advantage of the EH approach is more evident in Figure 11(d), where the objects are relatively large. However, note that at the the higher grid levels, the estimation accuracy of EH decreases sharply. The reason for the performance degradation is that as the grid granularity increases, the heights and widths of the objects (or *pieces* of the objects) becomes comparable to the heights and widths of the grid cell. As we discussed in Section 5.2, neither MP model nor GH model handles this case very well.

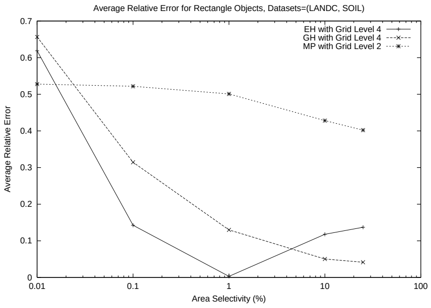
6.2.2 Impact of Area Selectivity

Based on the results from previous section, we identify the "optimal" grid level of a dataset for each algorithm, and use this grid level to study the impact of different area selectivity on the estimation accuracy. The results of this experiment are shown in Figure 12. The Y-coordinate of a data point is the average relative estimation error computed from 20 randomly generated selection windows, and the X-coordinate is the area selectivity of these selection windows, ranging from 0.01% to 25%.

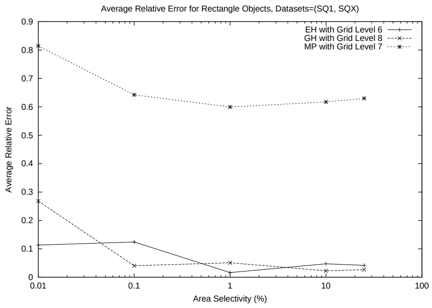
The general trend of the graph is that the estimation accuracy improves as the area selectivity increases. This is because with small selection windows, the number of the actual result is small, which leads to large *relative* errors.



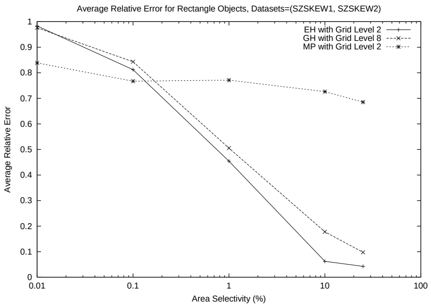
(a) LANDC ⋈ LANDO



(b) LANDC ⋈ SOIL



(c) SQ1 ⋈ SQX



(d) SZSKEW1 ⋈ SZSKEW2

Figure 12: Average Relative Error for Queries with Different Area Selectivity

In most cases, both EH and GH outperform MP by a large margin. Between EH and GH, EH performs significantly better at lower area selectivity. As we discussed in the previous section, EH estimates the number of intersection regions, while GH estimates the number of intersection points. When the selection window is small, a relatively large number of intersection points may fall outside the selection window, which decreases the estimation accuracy of GH.

7 Conclusion and Future Work

To estimate the selectivity of spatial joins, histogram based techniques must address two important issues: the objects that span multiple buckets and the probabilistic model that provides per-bucket estimations. The problem is further complicated when the spatial joins are constrained by selection windows, which has a significant impact on the design of the histogram as well as the accuracy of the probabilistic model.

In this paper, we propose the generalized Euler Histogram as a framework for estimating the selectivity of SJGS queries. This framework is designed to estimates the number of intersection regions, which ensures that for an SJGS query, the information needed for estimation is *inside* the selection window. More importantly, under this framework, different probabilistic models can be used to better capture the size and spatial distributions of the data objects.

Our current implementation of the generalized Euler histogram has validated the effectiveness of this framework. In the future, we plan to further improve the performance of this implementation by exploring alternative probabilistic models. We are also interested in extending the Euler Histogram approach to polygon datasets, which provides more accurate cost estimation for spatial operations that use aggressive filtering techniques [BA99, KR01].

References

[AN00] Ashraf Aboulnaga and Jeffrey F. Naughton. Accurate estimation of the cost of spatial selections. In *ICDE'00, Proceedings of the 16th International Conference on Data Engineering*, pages 123–134, March 2000.

[Ana96a] Wyoming Gap Analysis. *Land Cover for Wyoming.* University of Wyoming, Spatial Data and Visualization Center, December 1996.

[Ana96b] Wyoming Gap Analysis. *Land Ownership and Management for Wyoming.* University of Wyoming, Spatial Data and Visualization Center, December 1996.

[APR99] Swarup Acharya, Viswanath Poosala, and Sridhar Ramaswamy. Selectivity estimation in spatial databases. In *SIGMOD'99, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 13–24, June 1999.

[AYS01] Ning An, Zhen-Yu Yang, and Anand Sivasubramaniam. Selectivity estimation for spatial joins. In *ICDE'01, Proceedings of the 17th International Conference on Data Engineering*, pages 368–375, April 2001.

[BA99] Wael M. Badawy and Walid G. Aref. On local heuristics to speed up polygon-polygon intersection tests. In *ACM-GIS '99, Proceedings of the 7th International Symposium on Advances in Geographic Information Systems*, pages 97–102, 1999.

[BT98] R. Beigel and Egemen Tanin. The geometry of browsing. In *Proceedings of the Latin American Symposium on Theoretical Informatics, 1998, Brazil*, pages 331–340, 1998.

[FSTJ00] Christos Faloutsos, Bernhard Seeger, Agma J. M. Traina, and Caetano Traina Jr. Spatial join selectivity using power laws. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, volume 29, pages 177–188. ACM, 2000.

[JAS00] Ji Jin, Ning An, and Anand Sivasubramaniam. Analyzing range queries on spatial data. In *ICDE'00, Proceedings of the 16th International Conference on Data Engineering*, pages 525–534, March 2000.

[KR01] Ravi K. Kothuri and Siva Ravada. Efficient processing of large spatial queries using interior approximation. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD 2001)*, pages 404–421, 2001.

[MA99] Larry C. Munn and Christopher S. Arneson. *Draft 1:100,000-Scale Digital Soils Map of Wyoming.* University of Wyoming Agricultural Experiment Station, 1999.

[MP01] Nikos Mamoulis and Dimitris Papadias. Selectivity estimation of complex spatial queries. In *SSTD'01, Proceedings of the 7th International Symposium on Spatial and Temporal Databases*, July 2001.

[RSV01] Philippe Rigaux, Michel Scholl, and Agnès Voisard. *Spatial Databases with Applications to GIS*, chapter 1.3.1, page 14. Morgan Kaufmann Publishers, 2001.

[SAE01] Chengyu Sun, Divyakant Agrawal, and Amr El Abbadi. Exploring spatial datasets with histograms (full version). Technical report, Computer Science Department, University of California, Santa Barbara, 2001. http://www.cs.ucsb.edu/TRs/Docs/TRCS01-03.ps.

[SFGM93] Michael Stonebraker, James Frew, Kenn Gardels, and Jeff Meredith. The sequoia 2000 benchmark. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, pages 2–11, 1993.