Selectivity Estimation for Spatial Joins *

Ning An Zhen-Yu Yang Anand Sivasubramaniam
Dept. of Computer Science & Engineering
The Pennsylvania State University - University Park
{*an, zyang, anand*}@cse.psu.edu

Abstract

Spatial Joins are important and time consuming operations in spatial database management systems. It is crucial to be able to accurately estimate the performance of these operations so that one can derive efficient query execution plans, and even develop/refine data structures to improve their performance. While estimation techniques for analyzing the performance of other operations, such as range queries, on spatial data has come under scrutiny, the problem of estimating selectivity for spatial joins has been little explored. The limited forays into this area have used parametric techniques, which are largely restrictive on the data sets that they can be used for since they tend to make simplifying assumptions about the nature of the datasets to be joined. Sampling and histogram based techniques, on the other hand, are much less restrictive. However, there has been no prior attempt at understanding the accuracy of sampling techniques, or developing histogram based techniques to estimate the selectivity of spatial joins. Apart from extensively evaluating the accuracy of sampling techniques for the very first time, this paper presents two novel histogram based solutions for spatial join estimation. Using a wide spectrum of both real and synthetic datasets, it is shown that one of our proposed schemes, called Geometric Histograms (GH), *can accurately quantify the selectivity of spatial joins.*

1. Introduction

Spatial Database Management Systems (SDBMS) [22] need to provide a range of specialized and optimized spatial operations, such as spatial selection, nearest neighbor query and spatial join. Of these operations, spatial joins are particularly important because they are not only commonly used, but can also serve as building blocks for more complex spatial predicates. Spatial joins also present interesting challenges because of their high CPU and I/O costs.

A spatial join finds pairs of objects (from different datasets) that meet a given spatial predicate, such as intersection/overlap, containment, etc. For example, the query "find all the major highways in Pennsylvania that cross a major river" can be answered by performing a spatial join on the highway and river datasets of Pennsylvania. In SDBMS, a spatial data object is typically abstracted/represented by its Minimum Bounding Rectangle (MBR), which is the smallest axis-parallel rectangle that fully contains this spatial object. Using MBRs, spatial joins are performed in two steps [19]: the filter step and the refinement step. The filter step retrieves all Minimum Bounding Rectangles (MBRs) that satisfy the given spatial predicate. The refinement step then examines the exact geometry of the pairs produced by the filter step to discard any false hits. Although the refinement step is an important issue, most prior research (as is this paper) has focused on the filter step.

A good deal of research [7, 20, 13, 16, 3] has been done on optimizing the filter step of spatial join processing. However, there is another important problem related to spatial joins: *How do we predict the performance (selectivity in particular) of spatial joins?* The spatial join selectivity of two datasets is the ratio of the resultant size of the spatial join to the size of the Cartesian product of both participants. As most prior research, this work considers only the filter step of the spatial join, and we thus deal only with two sets of axis-parallel rectangles (in a 2-D space). The spatial predicate for the join in this paper extracts pairs of intersecting MBRs from the two datasets. Even with this simplication, accurately estimating the spatial join selectivity poses problems because (a) data items are located in a multidimensional space (instead of a single dimension in the traditional RDBMS), and (b) size of the spatial objects can vary significantly.

Selectivity estimation is crucial in a query optimizer for choosing a good execution plan for a given query. Selectivity estimates of spatial joins can themselves be used as responses to specialized user queries that are seeking approximate figures. For instance, finding the approximate number of bridges in a given spatial extent may simply be satisfied by doing a join selectivity estimation between the streets and rivers datasets for that extent (and may not necessitate performing the actual join). Finally, spatial join selectivity can also be used for evaluating the correlation between datasets [8].

The utility of selectivity estimation for spatial operations is widely recognized [22]. While there have been a large number of forays into this topic in the context of range queries [15, 24, 5, 27, 26, 14], the problem of selectivity estimation for spatial joins has been little explored. There are two prior studies, [12] and [25], that have extended

*This research has been supported in part by NSF Career Award MIPS-9701475, NSF grants CCR-9988164, CCR-9900701, EPA grant R825195-01-0, and NSF equipment grant EIA-9818327.

1063-6382/01 $10.00 © 2001 IEEE

prior analytical models for range query costs, to estimate the I/O performance of joins using R-trees. To our knowledge, there have been very few attempts [2, 6, 8] at selectivity estimation for spatial joins. Taking one dataset as the source of query windows, and the other as the underlying data, [2] simply applies the technique proposed in [15] for range query estimation, and sums these results to get a convenient closed form formula. Alternatively, [6] uses fractal concepts to estimate the selectivity of spatial self-join for point datasets. Along the same line, [8] uses a power law to model the distribution of pair-wise distance between two real multidimensional point datasets. Using this law, a fairly accurate selectivity estimation is derived for the spatial join of two point datasets.

Selectivity estimation techniques can be broadly categorized into three classes: parametric, sampling and histograms. Parametric techniques typically make some assumptions about the dataset to present convenient closed-form formulae for estimation, at little cost. For instance, [2] assumes that the data items are uniformly distributed in the two datasets to be joined, while [6] and [8] assume that the data items exhibit fractal behavior or obey a power law respectively. However, these assumptions restrict their applicability since real datasets may not necessarily adhere to such properties. Further, [6] and [8] can work only with point datasets. The other two classes of estimation techniques, sampling and histogram-based, try to draw sufficient information from the given dataset to predict query selectivity. As a result, they are applicable to a larger class of datasets than their parametric counterparts. Sampling techniques actually perform the query on a much smaller version of the dataset, called the sample , and use the results to project the selectivity on the entire dataset.

The difficulty in picking a representative sample with low overheads makes sampling somewhat undesirable. Histogram-based techniques, on the other hand, keep certain information for different regions of the spatial extent in an auxiliary data structure (histograms), and quickly consult this structure to find the selectivity when the query is given. The trick with histograms is in finding out what information to maintain and at what granularity, so that duplication across buckets of the histogram or the lack of information within each bucket does not significantly impact accuracy.

This paper intends to fill a crucial void in selectivity estimation of spatial joins by proposing and evaluating different sampling and histogram based techniques. While sampling techniques [10, 11, 4] have been used in estimation for conventional databases, less effort has been spent to investigate their usability in SDBMS: [18] dealt with techniques for obtaining random sample points of the query results and [28] intended to obtain approximate answers of aggregate queries using random sampling algorithms. This paper, on the other hand, studies three well-known sampling techniques to estimate the selectivity of spatial joins. In addition, two novel histogram based techniques are proposed. Using a diverse spectrum of real and synthetic datasets, that exhibit wide spatial distributions/patterns, these techniques are examined in terms of the estimation error and the estimation costs (both time and space), compared to performing the actual join.

It is shown that in most cases, picking samples randomly, with a sample size of 5-10% of the dataset, gives less than 10% errors at a overhead that is around 10% of the join time when the R-trees for the two datasets are not available. However, this is not a worthwhile option if the R-trees are available since the join itself is not as expensive. One of the undesirable properties of sampling is that the results are unstable i.e. it is highly dataset and sample dependent, and it is difficult to draw concrete conclusions.

On the other hand, one of the histogram based techniques that we propose in this paper, called the Geometric Histogram (GH) scheme, is shown to bring errors down to less than 5% with little overheads. This scheme uses extensive adjustments within and across buckets to avoid multiple and/or false counting of pairs in the join estimation. It is shown that both of our proposed histogram schemes can give much more accurate (and stable) results than the only known prior parametric technique for join selectivity estimation that has been discussed in [2].

The rest of the paper proceeds as follows. Sections 2 and 3 present the sampling and histogram based techniques for estimating spatial join selectivity. These techniques are then experimentally compared in section 4 using a wide range of datasets. Finally, Section 5 summarizes the contributions of this paper, and offers suggestions for future work.

2. Sampling Techniques

While sampling techniques have been used [10, 11, 4] to estimate the selectivity of equi-join, which is the counterpart of the spatial join in the relational DBMS, there has been few prior investigation, to our knowledge, of the applicability of these techniques to spatial data. In this study, we pick samples from both input datasets to be joined, and an R-tree [9] is then constructed for each of these samples. While one could try to directly perform a plane sweep algorithm [21] on the two samples, we have found that constructing an R-tree for the samples, then performing an R-tree join [7] is a better alternative, since even a small percentage of the datasets (which can be large) can result in a large number of data items to be joined. Suppose the sample sizes are $\alpha\%$ and $\beta\%$ of the the original datasets respectively, the estimated join selectivity is given by $\frac{R}{\alpha\% \times \beta\%}$, where R is the selectivity of the join on the samples. We consider the following three techniques to pick samples from the two datasets:

1. Regular Sampling (RS): If the sample size is n and the dataset size is N, RS generates a sample by taking every kth data item ($k = \lceil \frac{N}{n} \rceil$).
2. Random Sampling With Replacement (RSWR): Every data item of the given dataset has an equal probability of being selected, with a chosen data item potentially being picked more than once.
3. Sorted Sampling (SS): This follows the same procedure as RS, except that the input dataset is first sorted based on the Hilbert values [15] of the data items.

3. Histogram Based Techniques

The following subsections present two histogram based techniques to estimate spatial join selectivity. The common theme between these techniques is that an auxiliary data structure, histogram file, is constructed from the original dataset beforehand. The spatial extent is first gridded into

equi-sized cells with a number of vertical (2^h) and horizontal (2^h) lines, where h denotes the level of gridding. The histogram file stores the necessary information for each of the resulting 4^h cells. Later, when estimating a spatial join selectivity, these files for the two datasets (to be joined) are consulted. The following techniques differ in what information is kept in each cell.

3.1. Parametric Histogram (PH) Scheme

In this subsection, we first describe one prior parametric scheme [2], and see how it estimates the spatial join selectivity. A simple and straightforward extension is then proposed to overcome its shortcoming.

3.1.1. Prior Approach. Assuming that both range queries and data are uniformly distributed over the entire spatial extent, Kamel and Faloutsos [15] developed an analytical formula to evaluate the average response time for a range query. This was later extended to estimate the selectivity of spatial joins [2]. The basic idea is to consider one data set as the underlying database and the other as a source for query windows. The sum of the estimated range query selectivities would then give an estimation of the spatial join selectivity.

Suppose we have the following parameters for dataset DS_k:

- A: the area of the entire given spatial extent.
- N_k: the number of all data items in the dataset DS_k .
- C_k: the data coverage, i.e. the ratio of the sum of the areas of each data item in the dataset DS_k to A.
- W_k: the average width of all data items in the dataset DS_k.
- H_k: the average height of all data items in the dataset DS_k.

Then, the selectivity of the spatial join between datasets DS_1 and DS_2 is estimated in [2] as:

$$Size_{1_2} = N_1 \times C_2 + C_1 \times N_2 + N_1 \times N_2 \times \frac{W_1 \times H_2 + W_2 \times H_1}{A} \quad (1)$$

$$Selectivity_{1_2} = \frac{Size_{1_2}}{N_1 \times N_2} \quad (2)$$

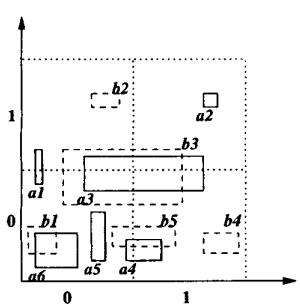


Figure 1. Extending PH

3.1.2. Proposed Extension (PH). While the parametric technique discussed in [2] incurs negligible time and space overheads (only requires computing Equation 1), the underlying assumption is that the data items are uniformly distributed in the spatial extent. Deviations from this assumption can result in significant errors in estimation as will be shown later in this paper. One way of fixing this problem is to grid the spatial extent into cells, with the hope that the uniformity assumption holds better within each cell. This leads us to propose a technique called Parametric Histogram (PH), wherein we maintain the necessary information (the parameters in Equation 1), for each grid cell in the histogram file. Selectivity estimation is then a sum of the selectivity over all the grid cells. While PH may appear straightforward, it has a drawback of multiple counting the intersections. For instance, in Figure1, MBRs $a3$ and $b3$ that span more than one cell actually intersect only once, but could be counted four times (in the four cells). Finer the gridding level (to better approximate uniformity within a cell), worse is the multiple counting problem. To alleviate this problem, our proposed PH scheme categories the MBRs from dataset DS_k that intersect with a cell(i,j) into two groups: $Cont_k(i,j)$ which contains MBRs that are fully contained within cell(i,j); and $Isect_k(i,j)$ which contains MBRs that intersect with cell(i,j), but are not fully contained within it. For dataset a in Figure 1, $Cont_a(0,0)$ includes MBRs $a5$ and $a6$ while $Isect_a(0,0)$ includes MBRs $a1$, $a3$ and $a4$.

For given datasets DS_1 and DS_2, the selectivity estimation for each cell(i,j) now needs to handle four cases: (a) intersection of $Cont_1$ and $Cont_2$; (b) intersection of $Cont_1$ and $Isect_2$; (c) intersection of $Isect_1$ and $Cont_2$; and (d) intersection of $Isect_1$ and $Isect_2$.

Parameters	Description
$AvgSpan_k$	average number of cells spanned by MBRs spanning cell boundaries
$Area_{cell}$	area of a cell.
$Num_k(i,j)$	number of MBRs that are fully contained in this cell (i.e. $Cont_k(i,j)$).
$Cov_k(i,j)$	ratio of the sum of areas of MBRs in $Cont_k(i,j)$ to $Area_{cell}$.
$Xavg_k$	average width of MBRs in $Cont_k(i,j)$.
$Yavg_k$	average height of MBRs in $Cont_k(i,j)$.
$Num'_k(i,j)$	number of MBRs that intersect this cell and cross cell boundaries (i.e. $Isect_k(i,j)$).
$Cov'_k(i,j)$	ratio of the sum of intersecting areas of MBRs in $Isect_k(i,j)$ with cell(i,j), to $Area_{cell}$.
$Xavg'_k(i,j)$	average width of **intersections** of MBRs in $Isect_k(i,j)$ with cell(i,j).
$Yavg'_k(i,j)$	average height of **intersections** of MBRs in $Isect_k(i,j)$ with cell(i,j).

Table 1. PH Parameters

Table 1 summarizes the parameters that are used to implement the PH technique for a given dataset DS_k. Note that except for the first two (which are for the entire dataset), the other parameters are maintained for each cell. The estimation for the above four cases (S_a, S_b, S_c, S_d) can then be calculated using these parameters as follows (directly drawn from Equation 1):

$$S_a(i,j) = Num_1(i,j) \times Cov_2(i,j) + Cov_1(i,j) \times Num_2(i,j) + Num_1(i,j) \times Num_2(i,j) \times \frac{Xavg_1(i,j) \times Yavg_2(i,j) + Yavg_1(i,j) \times Xavg_2(i,j)}{Area_{cell}}$$

$$S_b(i,j) = Num_1(i,j) \times Cov'_2(i,j) + Cov_1(i,j) \times Num'_2(i,j) + Num_1(i,j) \times Num'_2(i,j) \times \frac{Xavg_1(i,j) \times Yavg'_2(i,j) + Yavg_1(i,j) \times Xavg'_2(i,j)}{Area_{cell}}$$

$$S_c(i,j) = Num'_1(i,j) \times Cov_2(i,j) + Cov'_1(i,j) \times Num_2(i,j) + Num'_1(i,j) \times Num_2(i,j) \times \frac{Xavg'_1(i,j) \times Yavg_2(i,j) + Yavg'_1(i,j) \times Xavg_2(i,j)}{Area_{cell}}$$

$$S_d(i,j) = Num'_1(i,j) \times Cov'_2(i,j) + Cov'_1(i,j) \times Num'_2(i,j) + Num'_1(i,j) \times Num'_2(i,j) \times \frac{Xavg'_1(i,j) \times Yavg'_2(i,j) + Yavg'_1(i,j) \times Xavg'_2(i,j)}{Area_{cell}}$$

The basic idea behind these formulations is to break up rectangles spanning multiple cells into smaller ones (at cell boundaries), and handle the resulting rectangles in their appropriate cells. Of the above four cases, only $S_d(i,j)$ may cause multiple counting when we sum up the values from all the cells (only this case deals with rectangles that intersect in multiple cells). To adjust for this multiple counting, we can divide $S_d(i,j)$ by the mean of $AvgSpan_1$ and $AvgSpan_2$ i.e. the number of cells in which a rectangle in one dataset is likely to intersect with one rectangle in the other dataset. It should be noted that this is only an approximation to lessen the impact of multiple counting of intersections, and is not exact. Finally, PH uses the following formula to estimate the required spatial join selectivity.

$$Size_{1_2} = \sum S_a(i,j) + \sum S_b(i,j) + \sum S_c(i,j) + \frac{\sum S_d(i,j)}{\frac{AvgSpan_1 + AvgSpan_2}{2}} \quad (3)$$

3.2. Geometric Histogram(GH) Scheme

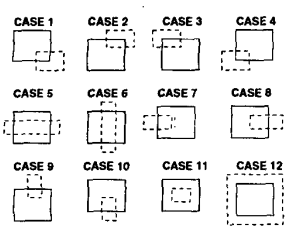


Figure 2. Intersections of Two Rectangles

This is a completely novel approach to spatial join selectivity estimation that is proposed in this paper. From Figure 2, one can observe that whenever two MBRs (rectangles) intersect with each other, the intersection is always another rectangle with four corners (let us call them *intersecting points*). Each intersecting point could be the result of one of the following two situations: (a) A corner point of one MBR falls inside another MBR (in Figure2, there are two such points in cases 1 through 4, two points in cases 7 through 10, and four points in cases 11 through 12); (b) A horizontal line of one MBR intersects with a vertical line of another MBR (in Figure2, there are two such points in cases 1 through 4, four points in cases 5 through 6, and two points in cases 7 through 10). If we can accurately estimate how many intersecting points exist between the two datasets, simply dividing this estimate by four will provide us the desired spatial join selectivity. To estimate the number of intersecting points between the two datasets, we propose a novel approach called the Geometric Histogram (GH) Scheme.

3.2.1. Basic GH. GH builds a histogram file for each dataset by gridding the spatial extent into cells (buckets) as discussed for PH. For an intuitive explanation of how GH

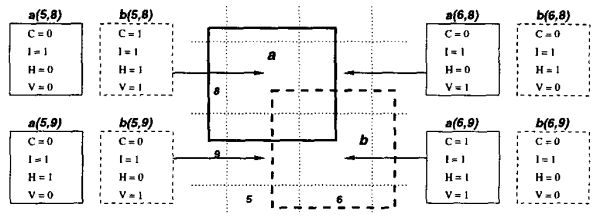


Figure 3. Example for Basic GH

works, let us say we record the following information for each grid cell (i,j): (a) how many vertical edges of MBRs pass through it ($V_k(i,j)$); (b) how many horizontal edges of MBRs pass through it ($H_k(i,j)$); (b) how many MBRs intersect it ($I_k(i,j)$); and (c) how many corner points of MBRs lie inside it ($C_k(i,j)$).

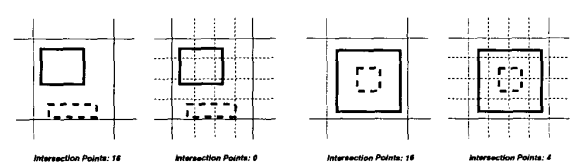
Then an estimate for the number of intersection points between datasets a and b can be made as follows:

$$N_{a_b} = \sum (C_a(i,j) \times I_b(i,j) + I_a(i,j) \times C_b(i,j) + V_a(i,j) \times H_b(i,j) + H_a(i,j) \times V_b(i,j)) \quad (4)$$

One can better understand this equation by examining the 16 cases of intersection in Figure 2 assuming that the gridding is done to such a fine granularity that the intersecting points between the two MBRs fall in different grid cells. In all these 16 cases, the above equation will correctly estimate four intersecting points (the first two terms calculate intersecting points corresponding to the sides of the two MBRs crossing each other, and the last two terms calculate intersecting points corresponding to a corner of one MBR falling within the other MBR). As an example, using equation 4, the number of intersecting points over the four grid cells that is shown in Figure 3 for MBRs a and b can be calculated to be 4.

We then divide the number of intersecting points by 4 to get the desired spatial join selectivity (1 in this case).

3.2.2. Revised GH. Equation 4 is based on the assumption that within a given cell, (a) every corner of the MBRs of one dataset falls inside all the MBRs of the other dataset which intersect this cell; and (b) every horizontal edge of the MBRs of one dataset intersecting this cell will intersect all the vertical edges of the MBRs of the other dataset intersecting this cell.



False Counting Multiple Counting

Figure 4. Inaccuracies in Basic GH

This can lead to errors that are illustrated in Figure 4 due to the granularity of gridding. As we go for a very

fine level of gridding, these errors would diminish, making the basic GH scheme more accurate. This is illustrated in Figure 4 which shows that the inaccuracies go away with a higher level of gridding. However, with a high level of gridding (number of grid cells grows exponentially), comes the high storage and processing costs, making it impractical. Instead, we propose to fix these inaccuracies by refining the basic GH scheme (with little additional overhead) as discussed below. The refinement is based on the assumption that data items are more or less uniformly distributed within each grid cell.

To facilitate our discussion, we use the notations in Table 2 representing the information GH will be needing for dataset DS_k in each grid cell (i, j).

Parameters	Description
$C_k(i,j)$	number of corner points that fall within cell (i,j).
$O_k(i,j)$	sum of the ratios of the intersection area (with cell (i,j) of MBRs to the cell area
$H_k(i,j)$	sum of the ratios of horizontal intersections (with cell (i,j)) of MBRs to the cell width
$V_k(i,j)$	sum of the ratios of vertical intersections (with cell (i,j)) of MBRs to the cell height

Table 2. GH Parameters

Suppose we want to estimate the selectivity of spatial join between dataset DS_1 and DS_2.

We will use MBRs a and b shown in Figure 5, which are from DS_1 and DS_2 respectively, to explain the basic idea of our approach when the estimation is done for the cell with width CW and height CH. The estimation of the intersecting points within a given cell is done as follows:

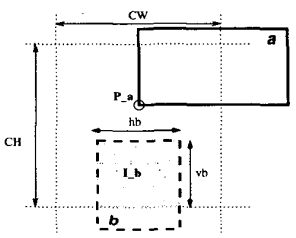


Figure 5. GH Adjustment for Corner Intersection Points

- Estimating corner intersecting points (such as P_a falling within b in Figure 5):

  The shaded area I_b represents the intersection of MBR b with the given cell, with the width and height of I_b being hb and vb respectively. Following the uniform distribution assumption, the probability of P_a falling in I_b is given by the ratio of the area of I_b (shaded area) to the area of the underlying cell, i.e. $\frac{hb \times vb}{CW \times CH}$. If DS_1 has N corner points inside this cell, statistically $N \times \frac{hb \times vb}{CW \times CH}$ of these points are likely to intersect I_b. Similarly estimating the intersections with the other MBRs of DS_2, gives $O_2(i,j) \times C_1(i,j)$ intersecting corner points of DS_1. Symmetrically there are $O_1(i,j) \times C_2(i,j)$ intersecting corner points of DS_2. Summing these two gives the total number of corner intersecting points in cell (i,j).

- Estimating vertical and horizontal line intersection:

  The probability that a vertical line of size v intersects with a horizontal line of size h inside a 2-dimensional space of $CW \times CH$, is given by $\frac{h \times v}{CH \times CV}$. The reader is referred to [1] for a proof of this observation. Adding this probability for all the vertical lines of DS_1 and horizontal lines of DS_2, we are likely to have $H_2(1,j) \times V_1(i,j)$ such intersecting points. Intuitively, we can get to this reasoning by going back to Equation 1 which estimates the number of intersecting rectangles in a 2-D space. If we simply set the areas C_1 and C_2 to zero, since we are dealing here with lines instead of rectangles, equation 1 degenerates to the formula used here. Symmetrically, we are likely to have $H_1(i,j) \times V_2(i,j)$ horizontal lines of DS_1 intersecting with vertical lines of DS_2 in cell (i,j).

Putting these arguments together, we estimate the number of intersecting points (IP)using the following equation:

$$IP = \sum (C_1(i,j) \times O_2(i,j) + C_2(i,j) \times O_1(i,j) + H_1(i,j) \times V_2(i,j) + H_2(i,j) \times V_1(i,j)) \quad (5)$$

This number is then divided by 4 to get the desired selectivity estimation.

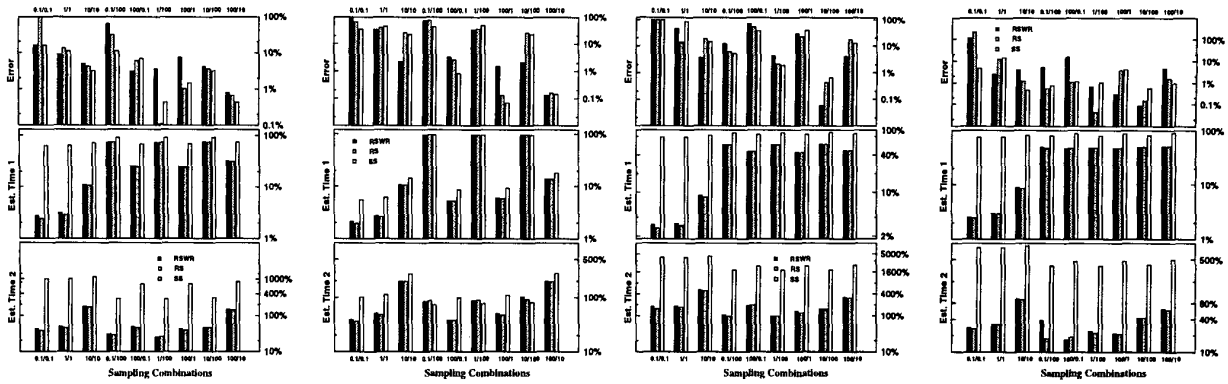
4. Evaluating the analysis techniques

In this section, we evaluate the accuracy and costs of the different sampling and histogram based techniques in estimating spatial join selectivity.

4.1. Datasets

To stress the pros and cons of the different schemes and their universal applicability, we have considered a wide spectrum of real and synthetic datasets. The selected datasets are quite diverse, and include both uniform and skewed spatial distributions. While the real datasets contain points, polylines and polygons, these are abstracted by their bounding boxes (MBRs) in our experiments, and the spatial join predicate is to find intersecting MBRs across the two datasets. Due to space limitations, we are not able to present the results for all the datasets. The reader is reffered to [1] for further information. In this paper, we present results for (a) **TS with TCB:** data of Iowa, Kansas, Missouri and Nebraska taken from the TIGER/Line(R) datasets [17] where TS contains the MBRs of 194,971 streams (polylines) and TCB contains the MBRs of 556,696 census blocks(polygons); (b) **CAS with CAR:** data of California taken from [17] where CAS contains the MBRs of 98,451 streams (polylines) and CAR contains 2,249,727 roads (polylines); (c) **SP with SPG:** data taken from the Sequoia benchmark [23] where SP contains the MBRs of 62,555 points and SPG contains 79,607 polygons; (d) **SCRC with SURA:** data synthetically generated in a 1×1 space where SCRC contains 100,000 rectangles clustered around (0.4,0.7) and SURA contains 100,000 rectangles uniformly distributed.

Using these datasets we consider different combinations of spatial joins that capture interesting and diverse facets: joins between datasets of different types such as TS with



(a) TS with TCB (b) CAS with CAR (c) SP with SPG (d) SCRC with SURA

Figure 6. Sampling Techniques Results

TCB (polylines with polygons), joins between datasets of the same number of data items, joins between datasets of unequal cardinalities, joins between datasets with different spatial skews, and joins between datasets with similar spatial skews.

4.2. Metrics of interest

To evaluate the pros and cons of the different techniques, we consider the following metrics:

- *Estimation Error*, which is the difference between that predicted by the techniques and the actual join selectivity normalized as a percentage with respect to the actual join selectivity.
- *Estimation Time*, which is the time to conduct the estimation relative to the time to perform the actual join using R-tree indices for the datasets
- *Space Cost*, which is the overhead in bytes for storing the required information for each technique, expressed as a percentage of the space required to maintain the R-trees for the actual datasets.
- *Building Time*, which is the time taken to construct the necessary information (histogram file for the histogram-based schemes, and samples for the sampling schemes), expressed as a percentage of the time taken to build the R-trees for the actual datasets.

A low estimation error and estimation time will be preferred. While building time is important if the target of the estimation is intermediary result(s) of a complex query, space cost is less important given the large amount of storage availability these days (as long as the storage requirements do not become comparable or exceed the dataset size itself). The statistics on the actual join of these datasets, together with the details on their R-trees can be found in [1].

4.3. Results for sampling techniques

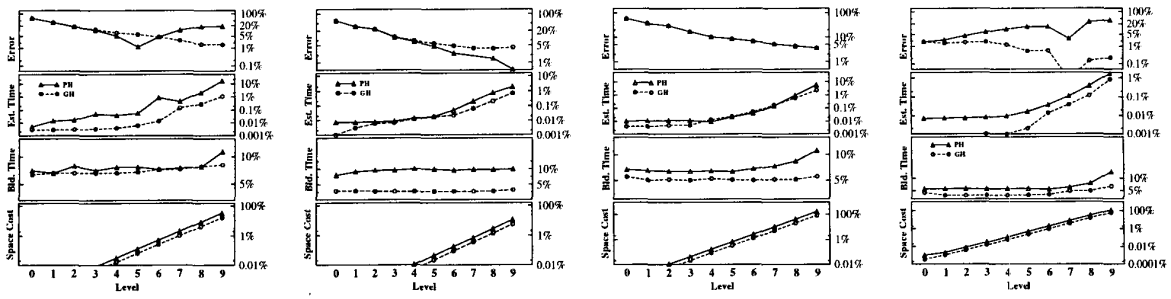
Figure 6 shows the results for the estimation of the spatial join selectivity with the various sampling schemes. All the bar graphs in these figures follow the same convention. The x-axis represents different sample size combinations. The first three sets of bars in these figures use samples (of sizes 0.1%, 1% and 10% of the datasets) from both datasets for the estimation. The fourth to ninth sets of bars use a sample from only one of the datasets, with the entire other dataset (shown as 100) being used. The individual bars within each set show the performance for the three sampling techniques discussed in Section 2.

All these graphs show the estimation error as defined in the previous subsection. The time cost is shown in two forms: *Est. Time 1* is the time overhead in selecting samples, building the R-trees from the samples and then performing the join, as a percentage of the time to do the actual join assuming the R-trees on the datasets are not available (i.e. they are built before the join is performed); and *Est. Time 2* is the same overhead assuming that R-trees are available, in which case the R-trees need not to be constructed for the original datasets. Obviously, Est. Time 1 is lower (as a relative percentage) compared to Est. Time 2. The space overheads are not explicitly shown here since they are apparent from the size (in percentage) of the samples that are chosen.

One can intuitively hypothesize that larger the sample, the more accurate the estimation. While this is an overall trend, we do find exceptions in some cases (such as RS for CAS with CAR when we go from 1/1 to 10/10, etc.). This is because the sampling idea is based on statistical arguments, and it is impossible to definitely say that a larger sample will necessarily give a more accurate estimate. However, it is fairly obvious from the graphs that larger samples incur higher time and space costs.

We find in Figure 6 that using all of one dataset and picking samples from only the other dataset does not pay off. The accuracy of this approach is not significantly better than picking a 10% sample from both datasets, and is in fact worse in many cases. Further, the time overheads are much worse than taking samples from both datasets if the R-trees on the two datasets are not available.

The other important consideration is the impact of the dataset size (or rather, the difference between the sizes of the two inputs to be joined) on the effectiveness of sampling. In general, we find that taking a smaller fraction from the larger dataset results in better estimation accuracy than



(a) TCB with TS (b) CAR with CAS (c) SPG with SP (d) SCRC with SURA

Figure 7. Histogram-based Techniques Results

taking the same fraction from the smaller dataset. This results in a much better statistical approximation of the two datasets. This also makes sense from the time cost viewpoint, since a larger fraction of the larger dataset incurs higher estimation overhead.

Between the three ways of picking samples, we find that Sorted Sampling (SS) is a poor choice. While its accuracy is not significantly better than the other two (in fact, it is worse in some cases), the sorting significantly adds to the time costs compared to the other two strategies. Regular (RS) and Random (RSWR) are more or less comparable, particularly for the synthetic datasets, where the data items are anyway generated randomly. With the real datasets, their relative performance really depends on the vagaries of the dataset (RSWR does better in two cases, and is comparable in the third). Hence, it is suggested that samples be generated randomly (RSWR) from the datasets.

In general, we find that if the R-trees are not available for the datasets, we can get the estimation error within 10% with sample sizes of 10% (i.e. 10/10), with time overheads that are also within 10% for random sampling (RSWR). This suggests that random sampling may be a viable option for spatial join estimation for intermediate steps/results (where the dataset is not previously available) of a long/complex query execution. When the R-trees are already available for the datasets, the results show that the estimation time costs (Est. Time 2) are much higher to get reasonable accuracy. However, one could argue that if R-trees are already available, then the samples for a dataset (and the R-trees on these samples) could also be made available beforehand. As shown in [1], with the availability of the R-trees on the samples, RSWR becomes once again a possible option with the estimation time cost being less than 10%.

4.4. Results for histogram based techniques

We next consider the two histogram-based techniques proposed in this paper. In figure 7, which shows the performance of the PH and GH schemes, the x-axis depicts the level of gridding (h, where 4^h is the resulting number of grid cells into which the spatial extent is histogrammed). The results are then shown in terms of the estimation error, estimation time, building time (for constructing the histograms), and the space overhead that have been described earlier.

We focus first on the results for PH. It should be noted that the PH results for $h = 0$ (the left most point in the curves) denotes the parametric model that has been originally proposed in [2], where the universe is assumed to be uniformly distributed and a simple formula is used to estimate spatial join selectivity based on this assumption. The other levels divide the space into equi-sized cells, and use the uniformity assumption within each such cell. There are two factors affecting the accuracy of the estimation as the number of levels is increased. We can expect better accuracy since a finer level of gridding will help better adhere to the uniformity assumption within each grid cell. However, finer gridding can result in data items spanning several grid cells, causing the estimation to multiple count (in several cells) the intersections (leading to an overestimation). Consequently, we expect the accuracy curves to first trend downward (the former factor is more significant) and then trending upward (the latter factor becomes more significant at higher h). This can be observed for TCB with TS join. Since the datasets for this join are clustered, the uniformity assumption hurts at lower levels. However, we find that we do not want to go beyond level 5, since the multiple counting starts hurting accuracy. In the joins for CAR with CAS and SPG with SP, the errors keep dropping even upto level 9. Since these datasets are highly skewed, the uniformity assumption is a severe restriction at lower levels. In the joins for SCRC with SURA, the uniformity assumption holds (SURA and SURB have been generated that way) causing the multiple counting factor to become more significant even at level 1. With increasing levels, the time and space costs go up as well. However, even at level 9, the estimation time takes less than 10% of the cost of performing the actual join, and the time for building the histogram file is also a rather small percentage of the time to build the R-trees. The sudden spike in building times at high levels is because the histogram file gets too large to fit in memory. It should be noted that the histogram file size is purely dependent on the level of gridding and not on the dataset itself. In summary, the PH scheme gives acceptable (10% errors) accuracy at level 5, with the time and space costs being negligible at this level of gridding.

Moving on to the GH scheme, we find the estimation errors monotonically decrease with the level of gridding. One can recall that this scheme attempts to avoid the double counting problem. As a result, it does not have the drawback that PH had with higher grid cells. Increasing the

gridding level makes the cells small enough so that the information within the cell is more accurately captured (false intersections are discounted). Consequently, the errors only decrease with gridding level. This is a nice property of GH which makes it somewhat more attractive than PH or any of the sampling schemes that are more unpredictable. The estimation time for GH is even lower than for PH. In fact, GH is very accurate (less than 5% errors) in all the four joins that are shown here, at level 7 (where the estimation time is around 1% or less). The space overhead for storing the histogram is typically 10% or lower for GH at this level.

In summary, GH is much more desirable than PH. Not only is the accuracy better for GH, but the results are much more stable as we increase the gridding level (PH requires us to find a good sweet spot for the gridding level). GH requires less space than PH (compare the information stored for the two schemes in Tables 1 and 2 ,and is also slightly less time consuming for each grid cell (compare Equations 3 and 5). These factors make GH a much better option than PH.

5. Concluding remarks and future work

Shekhar et al. [22] identify analysis of common spatial operations to be a crucial and daunting open problem for the success of SDBMS. This paper attacks the selectivity estimation of spatil joins by exploring the suitability of well-known sampling techniques and proposing two histogram-based techniques. One of the proposed histogram-based techniques, called the Geometric Histogram (GH) scheme, consistently brings error down to less than 5% with little overheads on various datasets.

In the future, we would like to develop analysis techniques for estimating selectivity and I/O costs for other spatial database operations, in addition to developing a SDBMS incorporating query optimizations based on these analysis techniques.

References

[1] N. An, Z. Yang, and A. Sivasubramaniam. Selectivity Estimation for Spatial Joins. Technical Report CSE-00-017, Dept. of Computer Science and Engineering, The Pennsylvania State University, July 2000.

[2] W. Aref and H. Samet. A Cost Model for Query Optimization Using R-Trees. In *Proceedings of ACM GIS*, pages 60–67, Gaithersburg, Maryland, November 1994.

[3] L. Arge et al. Scalable Sweeping-Based Spatial Join. In *Proceedings of VLDB*, pages 570–581, New York City, New York, 1998.

[4] J. S. B. Harangsri and A. Ngu. Selectivity estimation for joins using systematic sampling. In *Proceedings of International Workshop On Database And Expert System Applications*, pages 384–389, Toulouse, France, 1997.

[5] A. Belussi and C. Faloutsos. Estimating the Selectivity of Spatial Queries Using the 'Correlation' Fractal Dimension. In *Proceedings of VLDB*, pages 299–310, Zurich,Switzerland, 1995.

[6] A. Belussi and C. Faloutsos. Self-spacial join selectivity estimation using fractal concepts . *ACM Transactions on Information Systems*, 16(2):161–201, April 1998.

[7] T. Brinkhoff, H. Kriegel, and B. Seeger. Efficient Processing of Spatial Joins using R-trees. In *Proceedings of the SIGMOD*, pages 237–246, Washington, D. C., 1993.

[8] C. Faloutsos, B. Seeger, A. Traina, and C. Traina. Spatial Join Selectivity Using Power Laws. In *Proceedings of SIGMOD*, pages 177–188, Dallas, Texas, 2000.

[9] A. Gutman. R-trees: A Dynamic Index Structure for Spatial Searching. In *Proceedings of SIGMOD*, Boston, Massachusetts, 1984.

[10] P. J. Haas, J. F. Naughton, and A. N. Swami. On the relative cost of sampling for join selectivity estimation. In *Proceedings of PODS*, pages 14–24, Minneapolis, Minnesota, 1994.

[11] P. J. Haas and A. N. Swami. Sampling-based selectivity estimation for joins using augmented frequent value statistics. In *Proceedings of ICDE*, pages 522–531, Taipei, Taiwan, 1995.

[12] Y. Huang, N. Jing, and E. A. Rundensteiner. Spatial Join Using R-tree: Breadth-First Traversal With Global Optimizations. In *Proceedings of VLDB*, pages 396–405, Athens, Greece, 1997.

[13] Y. W. Huang, N. Jing, and E. A. Rundensteiner. A Cost Model for Estimating the Performance of Spatial Joins Using R-trees. In *Proceedings of Statistical and Scientific Database Management*, pages 30–38, Olympia, Washington, 1997.

[14] J. Jin, N. An, and A. Sivasubramaniam. Analyzing Range Queries on Spatial Data. In *Proceedings of ICDE*, pages 525–534, San Diego, California, March 2000.

[15] I. Kamel and C. Faloutsos. On Packing R-trees. In *Proceedings of CIKM*, pages 490–499, Washington D. C., 1993.

[16] M. L. Lo and C. V. Ravishankar. The Design and Implementation of Seeded Trees: An Efficient Method for Spatial Joins. *IEEE TKDE*, 10(1):136–151, 1998.

[17] U. S. B. of the Census. TIGER/Line(R) 1995 Data. http://www.esri.com/data/online/tiger/index.html. last visited: Feb. 10, 2000.

[18] F. Olken and D. Rotem. Sampling from Spatial Databases. In *Proceedings of ICDE*, pages 199–208, Vienna, Austria, 1993.

[19] J. A. Orenstein. Spatial Query Processing in an Object-Oriented Database System. In *Proceedings of SIGMOD*, pages 326–336, 1986.

[20] J. M. Patel and D. J. DeWitt. Partition Based Spatial Merge Join. In *Proceedings of SIGMOD*, pages 259–270, Quebec, Canada, 1996.

[21] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[22] S. Shekhar, S. Chawla, S. Ravada, et al. Spatial Databases - Accomplishments and Research Needs. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):45–55, 1999.

[23] M. Stonebraker et al. The Sequoia 2000 Benchmark. In *Proceedings of SIGMOD*, pages 2–11, Washington, D.C., 1993.

[24] Y. Theodoridi and D. Papadias. Range Queries Involving Spatial Relations: A Performance Analysis. In *Proceedings of COSIT '95*, pages 537–551, Semmering, Austria, 1995.

[25] Y. Theodoridis et al. Cost Models for Join Queries in Spatial Databases. In *Proceedings of ICDE*, pages 476–483, 1998.

[26] Y. Theodoridis et al. Efficient Cost Models for Spatial Queries Using R-Trees. *TKDE*, 12(1):19–32, 2000.

[27] Y. Theodoridis and T. Sellis. A Model for the Prediction of R-tree Performance. In *Proceedings of ACM PODS*, pages 161–171, Montreal, Canada, 1996.

[28] M. Vassilakopoulos and Y. Manolopoulos. On Sampling Region Data. *DKE*, 22(3):309–318, 1997.