An Interactive Framework for Raster Data Spatial Joins

Wan D. Bae
Department of Computer Science
University of Denver
wbae@cs.du.edu

Petr Vojtěchovský
Department of Mathematics
University of Denver
petr@math.du.edu

Shayma Alkobaisi
Department of Computer Science
University of Denver
salkobai@cs.du.edu

Scott T. Leutenegger
Department of Computer Science
University of Denver
leut@cs.du.edu

Seon Ho Kim
Department of Computer Science
University of Denver
seonkim@cs.du.edu

ABSTRACT

Many Geographic Information Systems (GIS) handle large geospatial datasets stored in raster representation. Spatial joins over raster data are important queries in GIS for data analysis and decision support. However, evaluating spatial joins can be very time intensive due to the size of these datasets. In this paper we propose a new interactive framework that allows users to get approximate answers in near instantaneous time, thus allowing for truly interactive data exploration. Our method utilizes two proposed statistical approaches: probabilistic join and sampling based join. Our probabilistic join method provides speedup of two orders of magnitude with no correctness guarantee, while our sampling based method provides an order of magnitude improvement over the full quad-tree join and also provides running confidence intervals. We propose a framework that combines the two approaches to allow end users to trade-off speed versus bounded accuracy. The two approaches are evaluated empirically with real and synthetic datasets.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Spatial databases and GIS

General Terms

Design, Performance

Keywords

raster spatial join, visualization, probabilistic join, sampling

1. INTRODUCTION

Geographic Information Systems (GIS) are used for storage and retrieval of large spatial datasets. Each dataset is usually called a layer. Example layers may be roads, rivers, land elevation, etc. Layers are related if they have the same geographic coordinates. Spatial joins between two or more data sets are one of the most common GIS queries for data analysis. An example might be finding all roads within 100 feet of rivers located at 1000 feet altitude or less. GIS users often want to visualize query results and being able to do so in an interactive fashion would greatly increase the utility of the GIS. Unfortunately, the large dataset size makes interactive spatial joins difficult.

Geospatial data is usually stored in one of two alternative data formats: raster (grid cells) and vector (point, line and polygon). In this paper we focus on raster data, where little research has been done on optimization or approximation techniques for spatial joins. Currently, performing spatial joins on raster data requires layers to be compared on a cell-by-cell basis. This spatial join process, referred to as *map overlay*, requires intensive computation time. To enable interactive queries, more efficient methods for dealing with raster data are needed.

GIS systems are often used to visualize results for the end user to assist in decision making processes. In many applications, obtaining an approximate join result in a reasonably short time is far more important than evaluating an exact join over a long time period. Fast response times are especially important for user-driven data exploration used in GIS. We believe GIS users should be given the chance to see which are the "interesting" dataset join pairs without having to wait to compute the actual full joins. In this paper we propose an interactive query processing framework for spatial join that enables GIS users to obtain an approximate "big picture" visualization of an answer in two orders of magnitude faster time than the time required for obtaining the exact answer.

Our general interactive framework works as follows. Users specify queries and get near instantaneous visualizations of the answer using our proposed probabilistic join method. These result visualizations are approximations with no guaranteed bound of correctness. For queries that had interesting results users can either use our proposed sampling algorithm to get a confidence bounded answer estimate, or, compute the full join. By allowing the user to get near instant approximate answers they are able to explore far greater numbers and sizes of datasets than previously possible. This increase ability does come at the cost of possibly

ACMGIS'07, November 7-9, 2007, Seattle, WA
Copyright 2007 ACM ISBN 978-1-59593-914-2/07/11 ...$5.00.

making a mistake and hence may not be appropriate for systems used in critical life decisions.

Our approach is based on two techniques:

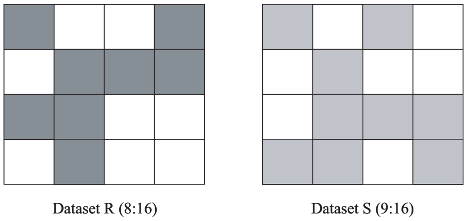


Figure 1: Raster cells of datasets R and S

- Probabilistic joins: The main idea is to calculate the join probability and the expected number of the joined cells of two raster datasets that have the same geographic coordinates. Data density (ratio of non-zero data cells to total data cells) in each node of a quad-tree of two datasets is used to calculate the join probability. Figure 1 shows an example of two datasets; R and S represented by a 4×4 raster grid. In this example, R has 8 non-zero data cells (density: 8/16) while S has 9 non-zero data cells (density: 9/16). Then R and S must intersect regardless of their shape and location. The ratio of non-zero data cells to the total number of data cells of each data set can be used in the calculation of the join probability and the expected number of joined cells.
- Sampling joins: Using quad-trees, overlapping blocks (sub-regions) are used to filter candidate pairs in order to speed up the joining process. Our sampling join approach is based on stratified random sampling from quad-trees and performing joins on the incremental samples to estimate the final answers of spatial joins with bounded confidence intervals.

Our proposed interactive framework combines the two proposed statistical approaches in order to speed up the process of obtaining estimations of the final join result in a reasonable time compared to the total time needed to perform a full join. Augmented quad-trees with non-zero data cells are used in the framework. We provide experimental results for both synthetic and real GIS datasets that demonstrate the efficacy of our approach comparing to full quad-tree joins. The speedup relative to a full quad-tree join increases as dataset size increases.

2. RELATED WORK

One common raster data spatial join technique is map overlay [10]. Raster overlay is straightforward when the input rasters have the same cell boundaries. The resulting raster can be obtained cell by cell from the originals using the relevant operations on the cell values. However, little research work has been done on map overlay optimization techniques.

Since GIS data can reach gigabytes and possibly terabytes in size, full layer overlays could take hours and even days to complete. This necessitates a need for approximation techniques. Most of the work on relational database join approximations can not be directly applied to spatial databases. In [3, 4] the authors presented an approximation technique of vector-based spatial joins. First they converted vector data to raster format and filtered the possible joined pairs using the Four Color Raster Signature in [3] and the Three Color Raster Signature in [4]. They combined progressive and conservative approximations [2] in a single approximation to speed up the filtering step in identifying intersecting polygons. Their proposed techniques motivated us to obtain the join probability of two raster datasets.

The quad-tree is a very popular hierarchical data structure for the representation of binary images and maps and it is commonly used in spatial databases [1, 11], i.e., indexing for query processing, and optimizing decomposition. Our work assumes datasets are indexed by quad-trees. Quad-tree based sampling has been proposed in [8, 11]. In [11], the authors presented the analysis of four different sampling methods proposed by [8]. They applied sampling algorithms to specific quad-tree implementations to obtain approximate aggregate query results. They proposed two models in order to analyze sampling costs while our sampling approach provides a faster approximation of the join result with a bounded confidence interval.

The idea of incremental sampling technique using R-trees to provide interactive spatial join processing was proposed in [12]. The authors proposed two R-tree based sampling methods that were used to incrementally refine the estimated join result while providing a bounded confidence interval. Their approach was applied for vector-based data rather than raster data. The proposed sampling method in this paper follows the same framework but using quad-trees instead of R-trees and with a more sophisticated sampling method.

Probabilistic query evaluation was studied for uncertain continuously changing data in relational databases [5]. In [6], the authors proposed probabilistic join over uncertain data. They provided techniques to answer queries that return results with probabilities exceeding a given threshold.

To the best of our knowledge, our work is the first attempt to apply probabilistic approaches to estimate raster-based spatial joins.

3. A FRAMEWORK FOR SPATIAL JOINS OVER RASTER DATA

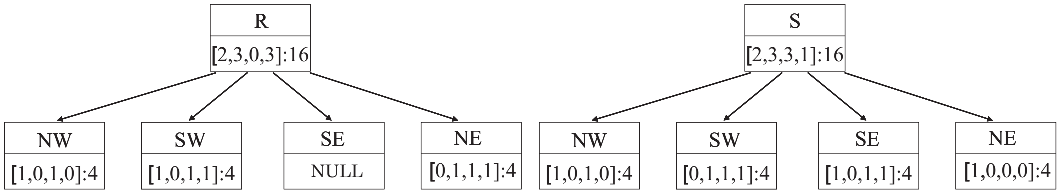
3.1 Augmented Quad-tree

Statistical methods are concerned with the estimations of parameters of the population in GIS. These approaches use information associated with the population, samples drawn from the population and distribution of the samples.

Augmented quad-tree data structure is used for PJ and $ISSJ$. Specifically, we augment nodes to include the total number of non-zero data cells of the subtree below. Our proposed statistical approaches use these augmented quad-trees for obtaining information associated with the population. Figure 2 (a) and (b) show two augmented quad-trees of the raster dataset examples in Figure 1. The nodes of the quad-trees are displayed in counter clock-wise order starting from the north-west. In our framework, all datasets are indexed by augmented quad-trees.

3.2 Probabilistic Joins vs. Random Sampling

In PJ, the augmented value (number of non-zero data cells) of each node of given two datasets is used to calculate



(a) Quad-tree of R (b) Quad-tree of S

Figure 2: Examples of augmented quad-trees of datasets R and S

the join probability and the expected number of joined data cells for each pair of subregions in the two joined datasets. PJ accesses nodes from the top to the bottom hence PJ is referred to as a top-down approach. PJ does not need to access all levels of quad-tree to calculate an estimate. It is mostly enough to access only a small number of top levels. Thus, it can greatly reduce time-consuming disk I/O operations in practice. The number of levels to be accessed is a system parameter. The greater number of levels is accessed, the more accurate the estimation can be. However, this would result in more I/Os. In the experiments, we set the number of levels to 4 resulting in only 64 nodes needed in memory, hence it is practical to store only required top level nodes of quad-trees for all joined data sets in memory. Although PJ provides no accuracy guarantee, our experimental results of synthetic and real data sets show the error bound is reasonably tight, e.g., a 9% error for 4^{th} level join (Section 5).

In $ISSJ$, stratified random sampling is used to estimate the final answer of spatial joins. An accuracy guarantee is provided in the form of error bound confidence intervals. In contrast to PJ, $ISSJ$ is performed on sampled leaf level data cells. Although far less number of I/Os are required in $ISSJ$ comparing to a full quad-tree join, obtaining a reasonable confidence interval requires a significantly greater number of I/Os comparing to PJ.

3.3 Framework Overview

We proposed a query processing framework that combines the proposed statistical approaches. The framework consists of three main processes: probabilistic joins, result visualizations and sampling joins. The main idea is to use PJ and a visualization technique to allow users to discover "interesting" data set pairs and areas for further data exploration. Once the user identifies interesting data sets, he/she can have the system perform $ISSJ$ in order to produce tighter running estimates of join results, or the user can have the system use the full quad-tree join to obtain the exact answer.

Figure 3 shows the overview of the framework, where two relations R and S are joined. 1) Probabilistic Joins (PJ): Given the user's interesting data sets, all higher level nodes (from level 0 to level 3 in our experiments) of the two data sets' quad-trees are loaded in memory. Then the join probability of each pair of the corresponding nodes is obtained from a look-up table. Since join probability is defined on continuous space, the system can use a lookup table for discrete values of join probability. 2) Visualization and user interface: Based on a visualized result of probabilistic joins, the user can identify "interesting join pairs". 3) Incremental Stratified Sampling Joins ($ISSJ$): $ISSJ$ starts incremental sampling process with the interesting pairs. Samples (non-zero cells) are randomly chosen from the outer relation R using stratified random sampling. Spatial joining on the corresponding cells of the inner data set is performed. The number of joined cells in each step is used to calculate a running estimate and a confidence interval for the final result. Finally, the calculated running estimate and confidence interval are combined with the intermediate result into a query result through visualization process. Then the query result is reported to the user. The user can stop the query process if the given confidence interval is sufficient or if the user sees satisfying trends from the visualized actual join locations (intermediate result), otherwise, each step of the process is repeated in an incremental manner to calculate new estimates until a desired confidence interval is achieved. Thus, the time to get join estimates needs to be compared to the time required for the full quad-tree join.

4. STATISTICAL JOIN APPROACHES

4.1 Probabilistic Join

Given a set X and two randomly chosen subsets A and B of X, what is the probability that $A \cap B \neq \emptyset$? Let us denote this probability by p. There is an easy answer in the finite case. Let $|X| = n$, $|A| = a$, $|B| = b$. Then $p = 1 - \frac{\binom{n-a}{b}}{\binom{n}{b}}$, since this is the probability that a randomly chosen b-element subset of X will not avoid a given a-element subset of X. But there is no reasonable answer in the infinite case, since we run into the well-known problems with (i) what is meant by "random" (the answer depends on how the experiment is conducted), (ii) measurability (how to determine the size of a set).

We therefore restrict our attention to subsets of special kind, and use the obtained answers as approximations to the (unsolvable) general case.

Theorem 4.1. *(Join Probability for intervals) Let $X = [0, 1]$, and let A, B be randomly chosen intervals in X of length a, b, respectively. Then, the probability p that $A \cap B \neq \emptyset$ depends only on a, b, and can be calculated by:*

$$p(a, b)_1 = \frac{1}{1-b} \int_0^{1-b} \frac{min\{x+b, 1-a\} - max\{0, x-a\}}{1-a} dx$$

Proof. Let A and B be $[a_l, a_h]$ and $[b_l, b_h]$, respectively, such that $\overline{a_l a_h} = |A| = a$ and $\overline{b_l b_h} = |B| = b$. If A and B are picked at random, then $a_l \in [0, 1-a]$ and $b_l \in [0, 1-b]$ (see Figure 4). Assuming that x is a random variable for the value of b_l, we have $x \in [0, 1-b]$. Then $p(a, b)$, the

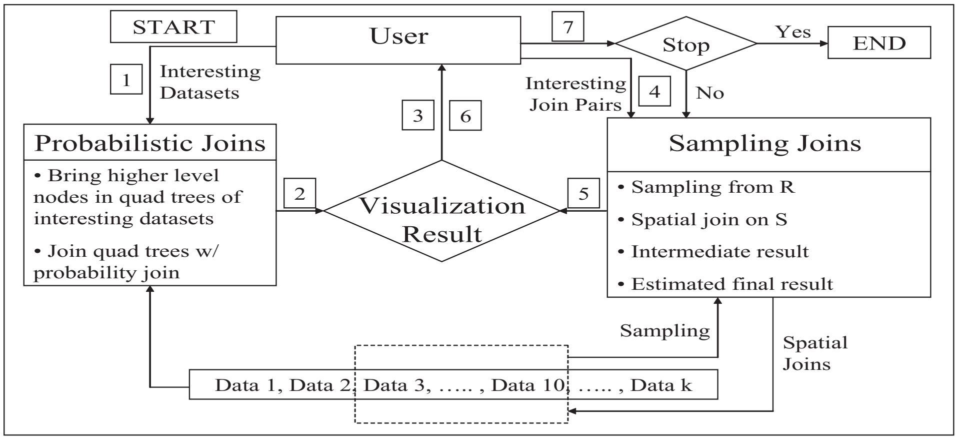


Figure 3: A framework for raster joins

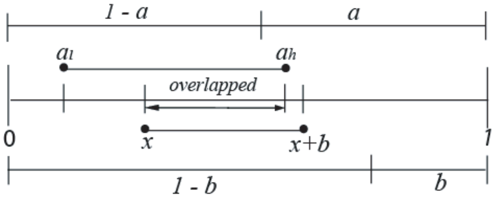


Figure 4: Join of two intervals

probability that $A \cap B \neq \emptyset$ (A intersects B), is as follows:

$$\begin{aligned} p(a,b)_1 &\equiv P((a_h \geq b_l) \wedge (a_l \leq b_h)) \\ &\equiv P((a_h \geq x) \wedge (a_l \leq min\{x+b, 1-a\})) \\ &\equiv P((a_l \geq x-a) \wedge (a_l \leq min\{x+b, 1-a\})) \\ &\equiv P(max\{x-a, 0\} \leq a_l \leq min\{x+b, 1-a\}) \end{aligned}$$

In order to have $p(a,b) \neq 0$, we need to pick a_l between $max\{x-a, 0\}$ and $min\{x+b, 1-a\}$ from the continuous space in which the range of x (b_l) is $[0, 1-b]$ and the range of a_l is $[0, 1-a]$. Then we have the following equation.

$$p(a,b)_1 = \frac{1}{(1-a)(1-b)} \int_0^{1-b} min\{x+b, 1-a\} - max\{0, x-a\} dx$$

□

Theorem 4.1 can now be generalized to any number of dimensions. The 2-dimensional case is as follows:

Let $X = [0,1]^2$, and let A, B be rectangles in X of area a, b, respectively. If the sides of A are of length a_1, $a_2 = a/a_1$ and the sides of B are of length b_1, $b_2 = b/b_1$, then we can use the 1-dimensional case to deduce that $P(A \cap B \neq \emptyset) = p(a_1, b_1) \cdot p(a_2, b_2)$. However, we do not know a_1 and b_1. All we know is that $a_1 \in [a, 1]$ (since the length of each side of A has to be at least a) and $b_1 \in [b, 1]$. We therefore conclude that:

$$p(a,b)_2 = \frac{1}{(1-a)(1-b)} \int_a^1 \int_b^1 p(a_1, b_1) \cdot p(\frac{a}{a_1}, \frac{b}{b_1}) da_1 db_1$$

It is now easy to see the general formula for two n-d prisms A, B in $X = [0,1]^n$ of volumes a, b, respectively. Let the lengths of sides of A and B be $(a_1, ..., a_n)$, $(b_1, ..., b_n)$, respectively. Then

$$p(a,b)_n = \frac{1}{(1-a)(1-b)} \int_a^1 \int_{\frac{a}{a_1}}^1 \cdots \int_{\frac{a}{a_1 \cdots a_{n-1}}}^1 \int_b^1 \int_{\frac{b}{b_1}}^1 \cdots$$
$$\cdots \int_{\frac{b}{b_1 \cdots b_{n-1}}}^1 u db_{n-1} \cdots db_1 da_{n-1} \cdots da_1,$$
$$\text{where } u = p(a_1, b_1) \cdot \cdot p(a_{n-1}, b_{n-1}) p(\frac{a}{(a_1 \cdot\cdot a_{n-1})}, \frac{b}{(b_1 \cdot\cdot b_{n-1})})$$

The expected overlapped length (area, volume) of A and B can be calculated using the conditional probability, since it is assumed that the two datasets are chosen independently:

$$P(A \cap B) = P(A) \cdot P(B)$$

The formulae for the join probability and the expected join numbers can be extended to more than two datasets joins.

4.2 Incremental Stratified Sampling Join

Sampling methods are used to estimate the final result from a subset (samples) of the data and to provide a bounded confidence interval. Query estimations and confidence intervals are statistically meaningful only if samples are retrieved at random. A weighted random sampling method, *Acceptance/Rejection* [8], is used in the *ISSJ* algorithm. We study stratified random sampling without replacement for raster data spatial joins. Each sampling is conducted in an incremental manner and the performance is evaluated with varying data sets and buffer sizes.

4.2.1 Stratified Random Sampling

Stratified random sampling is chosen because its property matches the property of quad-trees that provides systematic decomposition of a space with no overlaps between subregions. In stratified random sampling, the given region (population of all data cells) is divided into a number of non-overlapping subregions called strata. Then each stratum contains a set of raster data cells. Stratified random sampling can result in smaller error bounds on an estimation and can reduce the sampling cost [9].

In our algorithm, stratification is based on non-overlapping geometric forms such as rectangles (nodes at each level). We define the internal nodes of the quad-tree for a given level as strata, i.e., the second level nodes of quad-tree are used as strata in our experiments. We assume that the strata is pre-defined in our experiments. Algorithm 1 describes the $ISSJ$ algorithm.

Samples (non-zero cells) are then randomly chosen from each stratum by conducting simple random sampling. The sample size of each stratum n_i, i=1, ..k, is calculated for every sampling step, and it is proportional to the total number of non-zero cells within that stratum. Then the sampling size for a sampling step is $n_s = \sum_{i=1}^{k} n_i$. If the value of the chosen data cell is 1, searching the corresponding joined cell of the inner data set is performed in the quad-tree of the inner data set (line 15 of Algorithm 1). If the value of the corresponding cell is 1, then two data cell join. For each stratum, we obtain the number of joined cells, and this number is used to calculate the estimate and confidence interval for the corresponding stratum. The sum of the joined cells of each stratum is the current intermediate result, and the estimates and confidence intervals of all strata are combined for an estimate and a confidence interval of the final answer. The user can stop the query process if the given confidence interval is sufficient, otherwise the process repeats.

Algorithm 1 $ISSJ(R, S, ST)$

```
 1: $ST = \{ST_1, . . . , ST_k\}$; $ST$ is a set of strata
 2: $I_1, . . . , I_k \leftarrow 0$; $C_I \leftarrow 0$ {the current joined cells for
    stratum $i$; confidence interval}
 3: $n_s \leftarrow 0$; $n_{init} \leftarrow 30$ {the sample size for a sampling step;
    the initial incremental sample size for a sampling}
 4: $n_1, . \ , n_k \leftarrow 0$; $s_1, . \ , s_k \leftarrow 0$ {the sample size for
    stratum $i$; the incremental sample size for stratum $i$}
 5: repeat
 6:   compute $s_1, s_2, . . . , s_k$ for $ST_1, ST_2, .., ST_k$ using $n_{init}$
 7:   $S \leftarrow \sum_{i=0}^{k} s_i$; $n_s \leftarrow n_s + S$
 8:   for $i = 1$ to $k$ do
 9:     $n_i \leftarrow n_i + s_i$
10:     for $j = 1$ to $s_i$ do
11:       $L \leftarrow$ choose a leaf from $ST_i$ at random
12:       $c_r \leftarrow$ choose a non-zero cell from $L$ at random
13:       if cell $c_r$'s value is 1 then
14:         $P_r \leftarrow$ the center point of the chosen cell $c_r$
15:         $c_s \leftarrow$ findJoinedCell($S$, $P_r$)
16:         if cell $c_s$'s value is 1 then
17:           $I_i \leftarrow$ add 1
18:         end if
19:       end if
20:       remove $c_r$ from $L$
21:     end for
22:     remove $L$ from $ST_i$ if $L$ is empty
23:   end for
24:   $I \leftarrow \sum_{i=0}^{k} I_i$
25:   $C_I \leftarrow$ Compute a confidence interval w/all $I_i$ and $n_i$
26:   $EV \leftarrow$ Compute an estimate w/all $I_i$ and $n_i$
27:   report $EV$, $C_I$, and $I$
28: until $C_I$ is sufficient to the user or all $ST_i$ are empty
```

4.2.2 Estimates for Stratified Random Sampling

To provide bounds on the accuracy of our result, we incrementally calculate the current estimate with a confidence interval. The estimates and confidence intervals of $ISSJ$ are based on population proportion and *the Central Limit Theorem* (CLT) [7, 9]. We use the binomial probability distribution [9] for statistics of $ISSJ$. In $ISSJ$, the population is the non-zero cells of the outer relation R and $\hat{p}$ is the fraction of the elements in the sample that possess the characteristic of interest ("join" in our algorithm). Hence $\hat{p}$ is the fraction of cells in the sample that joins with the corresponding cell of the inner relation S. Confidence intervals depend on the size of samples and the distribution of the sample space (i.e., *Student t-distribution*).

Let N be the size of population (total number of non-zero cells of the outer datasets) and n_s be the sample size for a sampling step. If N_i is the number of non-zero cells in stratum i, and n_i is the sample size for stratum i, then $N = \sum_{i=1}^{k} N_i$, and $n_s = \sum_{i=1}^{k} n_i$, where k is the number of strata. Let I_i be the total number of cells that join the corresponding cells of S in stratum i. The following equations are used for a sampling step for $ISSJ$:
Estimator of the population proportion, where $\hat{p}_i = \frac{I_i}{n_i}$:

$$\hat{p} = \frac{1}{N}(N_1\hat{p_1} + N_2\hat{p_2} + ... + N_k\hat{p_k}) = \frac{1}{N}\sum_{i=1}^{k} N_i\hat{p_i} \qquad (1)$$

Estimate variance of $\hat{p}$:

$$\hat{V}(\hat{p}) \quad = \quad \frac{1}{N^2}\sum_{i=1}^{k} N_i^2 (\frac{N_i - n_i}{N_i})(\frac{\hat{p_i}\hat{q_i}}{n_i - 1}) \qquad (2)$$

Confidence interval:

$$E = t_c\sqrt{\hat{V}(\hat{p})} \qquad (3)$$

where t_c is the critical value for confidence level c taken from a Student t-distribution. Equations (1), (2) and (3) are valid for the incremental stratified sampling process. The proof of incremental equations can be found in our technical report [13].

5. EXPERIMENTS

In this section, we present experimental results of PJ and $ISSJ$ with both synthetic and real GIS datasets. The performances of two are compared with each other as well as with the full quad-tree join.

5.1 Data Sets and Experimental Methodology

In our experiments, we consider both synthetic and real data sets shown in Table 1. We generated four sets of uniformly distributed raster data and four sets of exponentially distributed (a mean of 0.3 and a standard deviation of 0.3) raster data. Our real data sets are from the 2001 and 2005 U.S. Geological Survey [14]: six datasets are chosen from Arizona, Colorado, Oregon and Wyoming in the US. These datasets are about minerals, stream sediments, water sediments, rocks, pluto sediments and unconsolidated sediments. Each dataset was converted into raster format. In Table 1, we present the total number of data cells (pixels), the total number of non-zero data cells and the data density for the synthetic and real datasets.

It is necessary that both the outer and inner datasets are indexed by augmented quad-trees and they have the same number of data cells as well as the same size of cells. Our experiments were conducted using the following parameters: Augmented quad-trees are implemented for PJ and $ISSJ$ while nonaugmented quad-trees are used for the full quad-tree join. The page size of the quad-tree was set to 4Kbytes, resulting in 100 nodes and 64 nodes for the non-augmented tree and augmented tree, respectively. Assuming an LRU replacement policy, we vary the buffer size: 5%,

	synthetic datasets								real datasets			
	uni1	uni2	uni3	uni4	exp1	exp2	exp3	exp4	AZ	CO	OR	WY
# total cells	65536	65536	262144	262144	65536	65536	262144	262144	65536	65536	65536	65536
# N.E. cells	17325	28365	39120	48298	14256	24736	36290	45231	6 datasets			
density	0.26	0.43	0.15	0.18	0.22	0.38	0.14	0.17	Mineral Resources			
description	uniformly distributed data				exponentially distributed data				from USGS			

Table 1: Synthetic and real datasets

10% and 20% of the size of one of the two relations. For all presented results, the estimates and the corresponding confidence intervals are shown with a 95% confidence level.

5.2 Experimental Evaluation

First we present the accuracy of join probability using the 1-dimensional formula (p_1) and 2-dimensional formula (p_2) discussed in Section 4. The total number of joins obtained by the 1-d and 2-d join probability were compared with the total number of actual joins. For discrete values of join probability, we created two lookup tables (20×20). Table 2 illustrates a portion of 2-d join probabilities from the lookup table used in the experiments. We randomly selected two corresponding nodes from the quad-trees of two real datasets. We checked the occupancy rates (non-zero data cells/total data cells) in the two chosen nodes and obtained the 1-d and 2-d join probabilities from the lookup tables. Then the expected numbers of joins were calculated. We repeated this process for varying size of sample pairs: 5%, 10%, 20% and 50% of the total quad-tree nodes. We ran the experiment 10000 times with each of the sample sizes and presented the average. In Table 3 we show the results of a join, unconsolidated sediments $\bowtie$ minerals in CO. The table entries are actual error values, thus, for example, an error of 0.1060 is a 10.60% error. Clearly, the 2-d join probability provides better approximation of the actual join.

P	0.2	0.4	0.6	0.8	1.0
0.2	0.7683	0.9277	0.9903	1.0	1.0
0.4	0.9277	0.9937	1.0	1.0	1.0
0.6	0.9903	1.0	1.0	1.0	1.0
0.8	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0

Table 2: Example of a 2-d lookup table

sample size	actual join	2-d (error)	1-d (error)
5 %	54	48 (0.1060)	39 (0.2778)
10 %	109	99 (0.0917)	78 (0.2844)
20 %	218	197 (0.0963)	155 (0.2889)
50 %	545	494 (0.0936)	389 (0.2862)

Table 3: Join probability

To evaluate the quality of the "big picture" visualization obtained by PJ, we calculated the expected number of joins using the 4^{th} level tree nodes. When using the 4^{th} levels of two quad-trees, only 64 subregions are joined. As a result, users can obtain the approximate result visualization in near instantaneous time with a truly interactive manner. We present results showing the difference between the PJ method and the full quad-tree join method (see Table 4). For the real datasets we compared PJ and $ISSJ$ for all 15 possible pairwise joins of the 6 datasets. We grouped the synthetic datasets into two: group 1 (uni1, uni2, exp1, exp2) and group 2 (uni3, uni4, exp3, exp4). We computed all possible 6 pairwise joins of each of the two groups. In Table 4, we present the average differences in the join density. The minimum and maximum of maximum difference, and the average maximum difference are also presented. Finally we calculated the average error in the expected number of joins of all the pairwise joins. As can be seen, PJ is reasonably accurate in all the cases of both real and synthetic datasets. With real data sets, PJ resulted in less accuracy due to the scattered clusters found in the datasets. As shown later in Figure 6, for the data we explored, these modest inaccuracies have little effect on the overall visual join-result appearance.

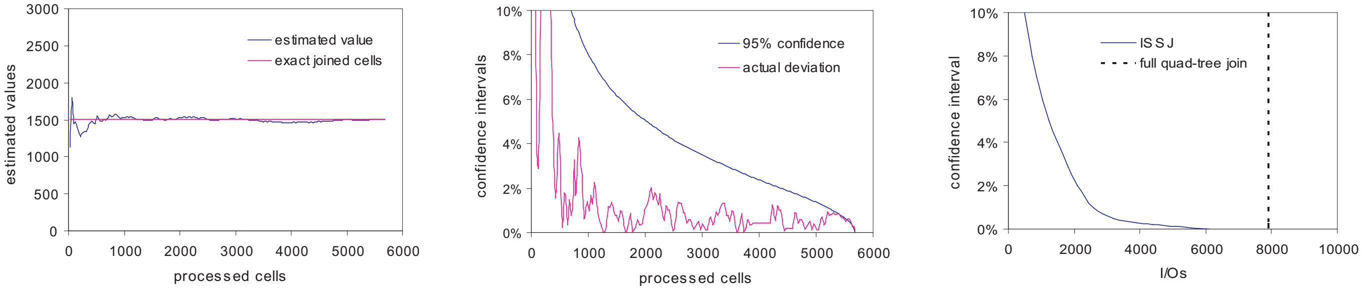
Next, we present the performance of $ISSJ$ compared to the augmented full quad-tree join. Figure 5 shows the result using the real datasets (minerals $\bowtie$ unconsolidated sediments from Colorado). The estimates and confidence intervals are plotted versus the number of samples (non-zero data cells) processed as well as the exact answer. Figure 5 (a) shows the estimated values of the final joins calculated by $ISSJ$. Figure 5 (b) shows how fast the confidence intervals converge. By showing the deviations from the actual joins, we demonstrate that $ISSJ$ provides good estimates of the final answer. In Figure 5 (c), we showed how fast an accurate estimation could be calculated compared to the time required for the full quad-tree join. For example, it took about 1900 I/Os to reach an estimate with a 5% confidence interval while 8,000 I/Os were required for the exact answer obtained by the full quad-tree join.

We next show how accurately the proposed approaches provide a "big picture" of the actual join. Figure 6 (a), (b) and (c) show three datasets for the state of Colorado: unconsolidated sediments (P), minerals (Q) and water sediments (S). The results of PJ and $ISSJ$ for $P \bowtie Q$ and $Q \bowtie S$ are presented as well as that of the actual join. The result from left to right corresponds to: $ISSJ$ with a 10% confidence interval (d), $ISSJ$ with a 5% confidence interval (e), actual joins (f) and finally PJ of the 4^{th} level nodes (g). PJ and $ISSJ$ with a 5% confidence interval provided a reasonably accurate approximation of the actual join.

In Figure 7 we present I/O comparisons between PJ and $ISSJ$ with varying the confidence intervals, as well as with the full nonaugmented quad-tree join (QT). All possible pairwise joins from the six datasets of CO and AZ were run and the number of I/Os were plotted for buffer sizes of 5%, 10% and 20% of the size of one dataset quad-tree. We plot the average total number of I/Os of each method averaged

join datasets	real datasets				synthetic datasets	
	AZ	CO	OR	WY	group1	group2
average diff.	0.0060	0.0087	0.0049	0.0058	0.0032	0.0024
minimum of max. diff.	0.0047	0.0038	0.0045	0.0014	0.0018	0.0015
maximum of max. diff.	0.1208	0.0973	0.0849	0.1143	0.0410	0.0312
average max. diff.	0.0329	0.0237	0.0214	0.0199	0.0201	0.0182
average error of estimates	0.1105	0.0729	0.0629	0.0904	0.0324	0.0229

Table 4: Join density differences of probabilistic joins from actual joins (4^{th} level)

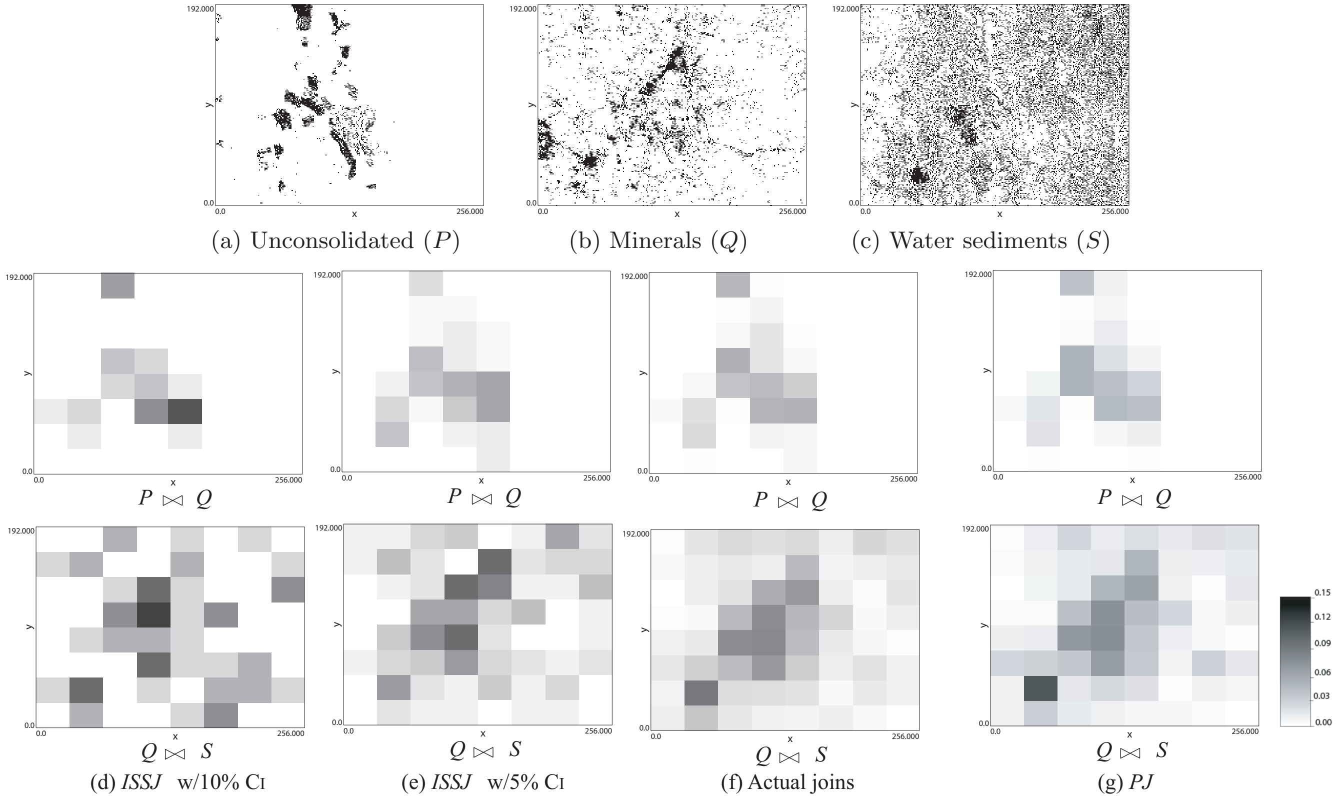


(a) Estimates

(b) Confidence

(c) I/Os

Figure 5: Estimates, confidence intervals and I/Os of $ISSJ$: unconsolidated sediment $\bowtie$ mineral in CO



(a) Unconsolidated (P)

(b) Minerals (Q)

(c) Water sediments (S)

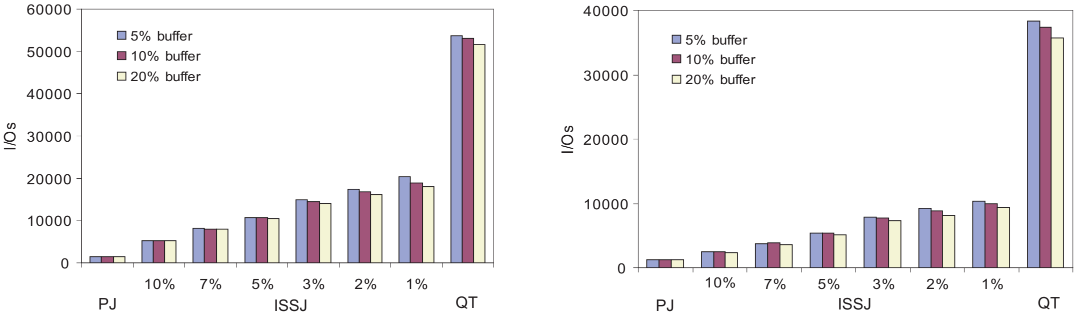
(d) *ISSJ* w/10% CI

(e) *ISSJ* w/5% CI

(f) Actual joins

(g) *PJ*

Figure 6: Expected number of joins: $ISSJ$ vs. PJ vs. Actual joins for real datasets in CO



(a) Colorado

(b) Arizona

Figure 7: Number of I/Os of PJ, $ISSJ$ and the full quad-tree join

over all 15 pairwise joins. The results for PJ are on the left, then $ISSJ$for confidence interval bounds of 10, 7, 5, 3, 2 and 1%, and finally the results for the full quad-tree join on the right. Note that the performance difference varying buffer sizes is very small since there is few re-visiting of the leaf nodes hence little opportunity to benefit from buffer caching.

The PJ method resulted in up to two orders of magnitude less I/Os than QT for both datasets. The $ISSJ$ algorithm obtained a very reasonable confidence interval (e.g. 5%) with far less I/Os compared to QT. PJ is significantly faster than the $ISSJ$ algorithm, but does not provide correctness bounds. However, PJ does provide a good overall picture for the data explored even though there is no guarantee of the quality of the estimate.

6. CONCLUSIONS AND FUTURE WORK

Due to the large dataset size, spatial joins of GIS data may take unreasonably long time to complete. The traditional map overlay joining method does not provide any idea of how the final result will look like until the join is completed. Hence, to enable an interactive data exploration, it is essential to allow a user to get a fast estimation, ideally a "big picture" visualization, of the join result. User comfort in using approximations can be increased by a method that also provides a confidence interval bound on the estimate.

In this paper, we proposed two statistical approaches for estimating spatial joins on quad-tree indexed raster data, namely, Probabilistic Joins (PJ) and Incremental Stratified Sampling Joins ($ISSJ$). We proposed a framework that combines two statistical approaches to allow fast interactive data explorations and the opportunity for the user to then drill down with full spatial joins if desired. Experimental evaluations on real and synthetic datasets showed that our proposed PJ method resulted in reasonably accurate results with near zero response time. Our $ISSJ$ method, while not as fast as PJ, provides results with bounded confidence intervals up to an order of magnitude faster than full quad-tree join. Our framework can be used to build an end-user query visualization tool that allows true interactive exploration of large raster based GIS databases.

In the future we plan to expand the PJ method for estimating the overlapping area of vector data (polygon) and integrating spatial joins between raster and vector data.

7. REFERENCES

[1] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, MA, 1990.

[2] T. Brinkhoff, H. P. Kriegel, and R. Schneider. Comparison of approximations of complex objects used for approximation-based query processing in spatial database systems. In *Proceedings of ICDE*, pages 40–49, 1993.

[3] G. Zimbrão and J. M. de Souza. A raster approximation for the processing of spatial joins. In *Proceedings of VLDB*, pages 558–569, 1998.

[4] L. G. Azevedo, R. H. Güting, R. B. Rodrigues, G. Zimbrão, and J. M. de Souza. Filtering with raster signatures. In *Proceedings of ACM GIS*, pages 187–194, 2006.

[5] R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proceedings of ACM SIGMOD*, pages 551–562, 2003.

[6] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. Vitter. Efficient join processing over uncertain data. In *Proceedings of CIKM*, pages 738–747, 2006.

[7] P. J. Hass. Large-sample and deterministic confidence intervals for online aggregation. In *Proceedings of SSDM*, pages 51–63, 1997.

[8] F. Olken. *Random Sampling from Databases*. PhD thesis, University of California at Berkeley, 1993.

[9] R. J. Serfling. *Basic Statistics for Business and Economics*. McGraw-Hill, 2002.

[10] H. Tveite. *Data Modeling and Database Requirements for Geographical Data*. PhD thesis, University of Norway, 1997.

[11] M. Vassilakopoulos and Y. Manolopoulos. On sampling regional data. *Data and Knowledge Engineering*, 22:309–318, 1997.

[12] W. D. Bae, S. Alkobaisi, and S. T. Leutenegger. An incremental refinining spatial join algorithm for estimating qeury results in GIS. In *Proceedings of DEXA*, pages 935–944, 2006.

[13] W. D. Bae, S. Alkobaisi, and S. T. Leutenegger. *IRSJ*: Incremental refining spatial joins for interactive queries in GIS. In *Technical Report DU-CS-07-10*. University of Denver, 2007.

[14] USGS. http://tin.er.usgs.gov/, 2001,2005.