Spatio-Temporal Histograms*

Hicham G. Elmongui, Mohamed F. Mokbel, and Walid G. Aref

Department of Computer Science, Purdue University, West Lafayette, IN, USA,
{elmongui,mokbel,aref}@cs.purdue.edu

Abstract. This paper presents a framework for building and continuously maintaining spatio-temporal histograms (ST-Histograms, for short). ST-Histograms are used for selectivity estimation of continuous pipelined query operators. Unlike traditional histograms that examine and/or sample all incoming data tuples, ST-Histograms are built by monitoring the *actual* selectivities of the outstanding continuous queries. ST-Histograms have three main features: (1) The ST-Histograms are built with (almost) no overhead to the system. We use only feedback (i.e., the actual selectivity) from the existing continuous queries. (2) Rather than wasting system resources in maintaining accurate histograms for the whole spatial space, we only maintain accurate histograms for that part of the space that is relevant to the current existing queries. The rest of the space has less accurate histograms. (3) The ST-Histograms are equipped with a periodicity detection procedure that predicts the future execution of the continuous queries. Hence, the query processing engine can continuously adapt the continuous query pipeline to reflect this prediction. Experimental results based on a real implementation inside a data stream management system show a superior performance of ST-Histograms in terms of providing accurate operator selectivity estimations with no extra overhead.

1 Introduction

The rapid increase in spatio-temporal applications calls for new query processing and query optimization techniques to deal with both the spatial and temporal domains. Examples of these applications include location-aware services [34], traffic monitoring [37], and enhanced 911 service [1]. These applications have two main characteristics: (1) A highly dynamic environment where data from mobile objects (e.g., moving vehicles in road networks) are received continuously. (2) Queries in these spatio-temporal applications are mostly *continuous* (e.g., monitoring queries). Continuous queries require continuous evaluation as the query area and/or the data are continuously moving.

Most of the previous work on continuous spatio-temporal queries (e.g., see [19, 22, 25–27, 42, 44, 47, 48]) focus on developing out-of-the-box algorithms (i.e., algorithms built on top of database management systems (DBMSs)). Having out-of-the-box algorithms bypass completely the role of the query optimizer

* This work was supported in part by the National Science Foundation under Grants IIS-0093116, IIS-0209120, and 0010044-CCR.

in DBMSs, thus severely limiting the query performance. Recently, within the PLACE server (*Pervasive Location-Aware Computing Environments*) [35], more attention is given to embed the functionality of continuous query processing into existing database and data stream query engines. The main idea is to furnish existing query processors with a set of *spatio-temporal* operators that can be combined with traditional operators to support efficient execution of a wide variety of continuous spatio-temporal queries. Our previous work in PLACE widens the scope of research in continuous spatio-temporal queries to include system-oriented support for continuous spatio-temporal queries (e.g., query optimization, adaptive query processing, and query scalability [35]).

In this paper, we focus on query optimization for continuous spatio-temporal queries. In particular, we are concerned with two main functionalities for optimizing the execution of continuous spatio-temporal queries: (1) Building a new *optimal* query plan for each newly submitted continuous query, and (2) Continuously monitoring the performance of continuous queries to make sure that the original *optimal* query plan maintains its optimality. Once the query optimizer discovers that the original query plan become suboptimal, the query optimizer tunes the suboptimal plan to another optimal one. To support these functionalities, we propose to build and continuously maintain spatio-temporal histograms (ST-Histograms, for short). Instead of monitoring the whole spatial space as in traditional histograms, ST-Histograms are *query-driven* where they monitor only the spatial space that is covered by at least one outstanding continuous spatio-temporal query. ST-histograms continuously maintain *spatio-temporal selectivity* estimations that are used by the query optimizer to decide on the optimality of various candidate query execution plans.

The proposed ST-Histograms start by an initial estimate of the *spatio-temporal selectivity* of the underlying spatial space. The accuracy of the initial estimation is continuously enhanced based on monitoring the execution of the continuous outstanding spatio-temporal queries. One of the attractive features of an ST-Histogram is that its accuracy (and hence the efficiency of the query execution) increases with the increase in the number of outstanding continuous queries. Moreover, the ST-Histogram consults some data mining techniques for periodicity detection (e.g., [13]) to provide better *spatio-temporal selectivity* with less overhead. All the algorithms and ideas in this paper are implemented as part of the PLACE project [4, 35] currently being developed at Purdue University.

1.1 Motivation

Spatio-temporal databases provide the ideal infrastructure for keeping track of and answering continuous queries on moving objects. To find an execution plan for a continuous query, the query optimizer needs to know (estimates of) the selectivity of any range that a query covers.

The distribution of the moving objects change with time. For instance, many cars go to downtown from 9am to 5pm leaving the suburbs with fewer cars. At night, most of the cars park, and hence deregister from the database. Consequently the number of the cars in the database is less. Obviously, building a

```
SELECT M.ID
FROM MovingObjects M
WHERE M.Type = "Truck"
INSIDE Area A
```

Fig. 1. Query Q

histogram once and using it for a long period is not enough. We need to maintain a spatio-temporal histogram and to reflect the change in the objects distribution on the histogram.

Not only the density of the moving objects change with time, but also there is some kind of periodicity in their behavior (as time repeats itself). For example, many people travel on weekends, yielding more traffic on the highways on Friday and Sunday evenings than on other week-evenings. Also, lots of traffic and congestion occur during the rush hour everyday. By detecting such patterns in the distribution of the moving objects, we believe that we can enhance the selectivity estimation.

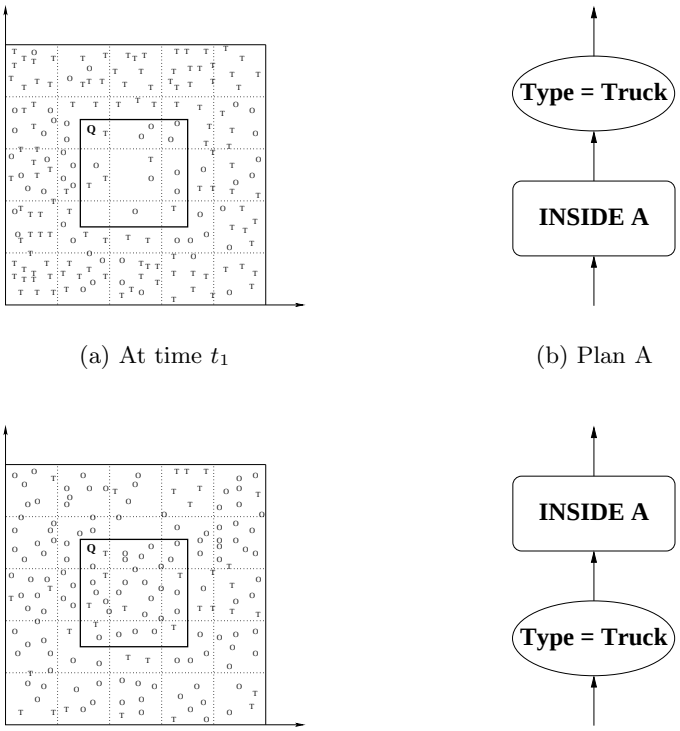
We illustrate the importance of having an ST-Histogram by the following example. Consider the query Q in Figure 1 that returns the ID of any truck whose location is inside an area A. The INSIDE query operator is proposed in [33] to check whether or not a moving object is in a certain range. Initially, at time t_1, the query optimizer finds that the selectivity of the INSIDE operator is less than the selectivity of the WHERE clause (Figure 2(a)). Thus the query optimizer picks up the query execution plan in Figure 2(b) to be used to answer the query. At time t_2, many vehicles enter the area A and this increases the selectivity of the INSIDE operator, and meanwhile the number of trucks decreases (Figure 2(c)). Using an ST-Histogram, the query optimizer is able to recognize that the current plan is suboptimal. The query optimizer calls for changing the current execution plan to the plan in Figure 2(d). Notice that query re-optimization is a non-trivial process, especially that the space of the possible execution plans is large.

1.2 Challenges and Paper Outline

The main challenges for ST-Histograms are the following:

- The large number of the moving objects, which is a computing challenge and a scalability challenge.
- Keeping the overhead of maintaining the ST-Histograms low and not to hurt the execution of the continuous queries.
- Having an accurate selectivity estimation with the frequent change in the data distribution over the time.

The rest of this paper is organized as follows. Section 2 highlights some of the related work in the areas of maintaining histograms and continuous queries. The architecture of our spatio-temporal histogram is given in Section 3. The role of the query executor is shown in Section 4. Section 5 introduces the histogram



(a) At time t_1 (b) Plan A

(c) At time t_2 (d) Plan B

Fig. 2. Moving vehicles (T = truck, O = other) on the spatial space and the corresponding query execution plan.

manager and the theory behind constructing and refining ST-Histograms. We discuss some query optimization issues in Section 6. In Section 7, we demonstrate by experiments the accuracy of ST-Histograms. Finally, Section 8 concludes the paper.

2 Related Work

Traditional histograms have been used extensively as a means for selectivity estimation in relational databases (e.g., [10, 11, 15, 17, 18, 24, 28, 29, 31, 32, 36, 38–41]). Currently, state-of-the-art histograms are *query-driven* (e.g. [2, 21, 23]). The main idea is to use a feedback from the query execution engine to estimate the data distribution. Thus, the cost of building the histograms is reduced where histograms are built during the regular process of query execution.

With the emergence of spatial applications, several approaches are proposed for the selectivity estimation of spatial operations (e.g., [3, 5–7, 14, 30, 43, 46]).

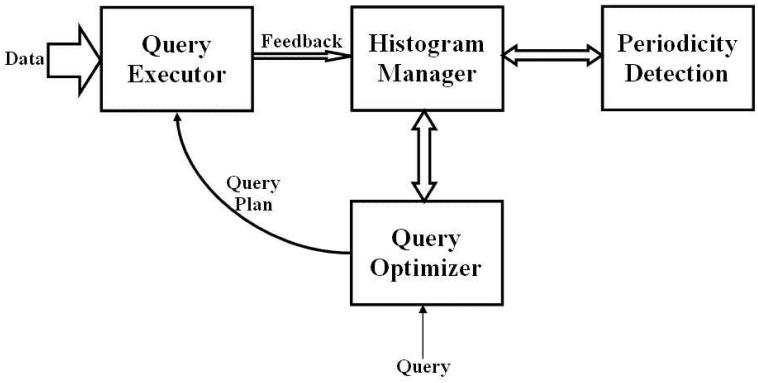


Fig. 3. The big picture

These approaches deal with the selectivity estimation in various data structures, e.g., selectivity estimation for quad trees [5], selectivity estimation for R-tree [6], and selectivity estimation for point data [8].

Recently, many research efforts focus on developing spatio-temporal histograms for various spatio-temporal operations. Spatio-temporal histograms are first proposed to provide selectivity estimation for predictive spatio-temporal queries [12]. The main focus is on one-dimensional moving objects. The selectivity estimation of multi-dimensional objects is computed by multiplying the selectivity estimations of each single dimension. Similar idea in the context of predictive spatio-temporal queries is introduced in [48]. However, the main focus is on multi-dimensional moving *rectangles*. Other work on selectivity estimation of spatio-temporal queries relies on duality transformation [20], the existence of a secondary index structure [20], or clustering approaches [50]. The state-of-the-art approach for spatio-temporal selectivity estimation is Venn sampling [49]. Venn sampling is a sampling technique that is not based on histograms where it aims to reduce the number of samples needed for *perfect* estimation. The main idea is allow each moving object to be aware of a set of some *pivot queries*. Moving objects update their locations and speed only when they start/cease to satisfy pivot queries.

3 Architecture

Figure 3 gives the big picture. Spatio-temporal queries are submitted to the query optimizer to generate adequate query execution plans. The query optimizer (e.g., System R [45] and Volcano [16]) picks the best execution plan based on the *total cost*. ST-Histograms provide the query optimizer with the selectivity estimates used in calculating the total cost. During the execution of a spatio-temporal query, feedbacks are sent to the histogram manager. These feedbacks

are typically the actual query selectivity; i.e., the fraction of the input data that is part of the query answer.

The histogram manager uses these feedbacks to refine the selectivity estimates online. The online refinement serves both the new incoming queries and the outstanding continuous queries. New incoming queries find a more accurate histogram, whereas the execution plan of outstanding continuous queries may be changed adaptively when the environment changes.

The ST-Histograms proposed in this paper consult an online periodicity mining technique (Section 6.1) to see if a periodic pattern appears in the selectivity of any region. Whenever such periodicity is detected, ST-Histograms take it into account to get more accurate selectivity estimates.

4 Query Executor

The spatial space is mapped with an NxN grid. When a moving object registers with a grid cell, the moving object sends periodic updates about its location. Hence, each grid cell is only aware of the objects inside it. Also, when a query registers with the system, the grid cells that overlap with the query are notified. Only when a change in the moving objects happens in those grid cells, the thread that executes this query is notified. In other words, each grid cell is aware of only the objects that are inside it and the queries that overlaps with it.

The query executor uses the plan provided by the optimizer to execute the query on the input data (Figure 3). Each INSIDE operator keeps track of the ratio of the number of its output tuples to the number of its input tuples. In PLACE [35], this ratio is part of the logic of the INSIDE operator. Thus this does not invoke substantial overhead on the executor. Periodically, such ratio is reported to the histogram manager as the selectivity of the INSIDE operator.

5 Histogram Manager

For streaming applications, we cannot afford storing the incoming data. The continuous query model does not allow for scanning the whole data in order to build the histogram. In fact, an ST-Histogram is built and refined progressively. We use feedback from the query result to update the spatio-temporal histogram online. Periodically, each operator reports the actual selectivity of its monitored range. These statistics are inherently computed in the operators. They do not impose additional overhead on the query executor.

Definition 1. ***Dark cell:*** *is a grid cell corresponding to a region with which no query overlaps.*

Definition 2. ***Lit cell:*** *is a grid cell corresponding to a region with which one or more queries overlap.*

Initially, the whole space is assumed to be dark, where darkness represents the unawareness of the selectivity. Queries act as spots of light. They light up a region with their feedback about the region's selectivity. We distribute this selectivity uniformly over the lit region. For the remaining dark regions, we consult the online periodicity mining technique (Section 6.1). If a region exhibits some kind of periodicity, its current selectivity can be estimated according to such periodicity. In other words, the periodic behavior of a region shades this region with little light when the corresponding grid cell is currently dark. The regions that neither fall inside the query regions nor exhibit any periodicity will stay dark. The selectivities of the remaining dark regions are estimated such that they complement the selectivities of those lit and shaded regions. This estimate is uniformly distributed over the ST-Histogram buckets that correspond to the dark regions.

The ST-Histogram is represented by a two-dimensional array. Each element of the array holds the selectivity of the corresponding histogram cell. We assume that a variable holds the total number of moving objects in the database.

5.1 Constructing the histogram

The ST-Histogram is grid-based of size NxN grid cells. The grid divides the universe uniformly into a number of disjoint cells. We denote the *current* view of the ST-Histogram with $\mathcal{H}$, where $\mathcal{H}[r, c]$ is the selectivity estimate of the grid cell $\mathcal{G}[r, c]$. Starting with all grid cells being dark, the selectivity estimate of each bucket is initialized uniformly according to Equation 1. With the successive feedbacks from the operators, better selectivity estimates are obtained due to a clearer view of the coverage area.

$$\mathcal{H}[r, c] = \frac{1}{N^2} \qquad \text{for all } r, c \in \{1, 2, \ldots, N\} \tag{1}$$

A query q is represented by a rectilinear rectangular region $\mathcal{R}_q$. Let $\mathcal{F}_q(r, c)$ be a scalar function that returns the fraction of the grid cell $\mathcal{G}[r, c]$ that is covered by $\mathcal{R}_q$. Hence, the selectivity estimate of a query q is calculated as in Equation 2. Similarly, let $\mathcal{F}_{\mathbf{dark}}(i, j)$ be the dark fraction of $\mathcal{G}[r, c]$. Thus the selectivity estimate of the whole dark area is calculated as in Equation 3.

$$\mathit{SelEst}(q) = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{H}[i, j] \mathcal{F}_q(i, j) \tag{2}$$

$$\mathit{SelEst}(\mathbf{dark}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{H}[i, j] \mathcal{F}_{\mathbf{dark}}(i, j) \tag{3}$$

5.2 Refining the histogram

When the histogram manager receives feedback from the query engine, it updates the histogram to reflect the newly reported statistics. Queries act as light

RefineHistogram$(\mathcal{H}, q, \mathcal{S})$
    $Diff = \mathcal{S} - SelEst(q)$;
    if $Diff > 0$
        $Diff = min(Diff, SelEst(\textbf{dark}))$;
    AddDiff$(\mathcal{H}, q, Diff)$;
    AddDiff$(\mathcal{H}, \textbf{dark}, -Diff)$;

AddDiff$(\mathcal{H}, q, Diff)$
    for $i = 1 : N$
        for $j = 1 : N$
            $\mathcal{H}(i, j) = \mathcal{H}[i, j] + \mathcal{R}_q(i, j) * Diff$;

Fig. 4. Procedure for refining the ST-Histogram

spots; they eliminate the darkness from a histogram region. The intensity of the light spot a query offers to a histogram region is proportional to the fraction of the histogram region illuminated by the query. When queries overlap, many light spots are directed on the overlapped histogram region. The more the light intensity of a histogram region, the better accuracy of the refinement of the selectivity estimate of this histogram region.

Definition 3. *The normalized rate of a query q for a grid cell $\mathcal{G}[r, c]$ is defined as the ratio between the selectivity estimation of the part of q that overlaps $\mathcal{G}[r, c]$ and the selectivity estimation of q. We denote this normalized rate by $\mathcal{R}_q(r, c)$.*

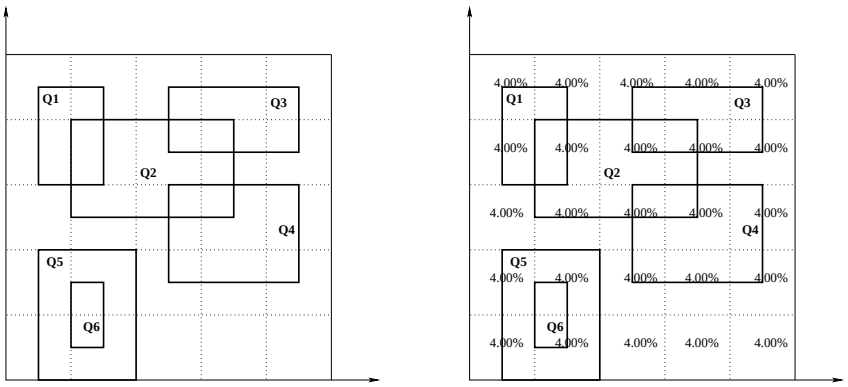
$$\mathcal{R}_q(r, c) = \frac{\mathcal{F}_q(r, c)\mathcal{H}[r, c]}{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathcal{F}_q(i, j)\mathcal{H}[i, j]} = \frac{\mathcal{F}_q(r, c)\mathcal{H}[r, c]}{SelEst(q)} \tag{4}$$

$$\mathcal{R}_q(r, c) = \frac{\mathcal{F}_q(r, c)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathcal{F}_q(i, j)} \qquad \text{When } SelEst(q) = 0 \tag{5}$$

The actual selectivity that a query reports is assumed to be distributed uniformly on the query range. Figure 4 gives the procedure to refine the histogram when a feedback from the query engine reports the actual selectivity $\mathcal{S}$ of a query q.

First, the grid cells overlapped by q will have their values changed according to the difference of $\mathcal{S}$ and the current selectivity estimate of q. Typically, this difference is distributed according to the normalized rate of q for each of these grid cells. Next, all the grid cells that have dark portions will be modified similarly to conform with the unity invariance of $\mathcal{H}$ ($\sum_{i=1}^{N}\sum_{j=1}^{N}\mathcal{H}[i, j] = 1$). Hence, the selectivity estimation of the dark portions is the upper bound for the difference when the difference is positive.

Example. We illustrate the histogram refinement by the example given in Figure 5(a). In this example, we have six continuous queries mapped to a 5x5



(a) Example of six continuous queries

(b) Initial ST-Histogram

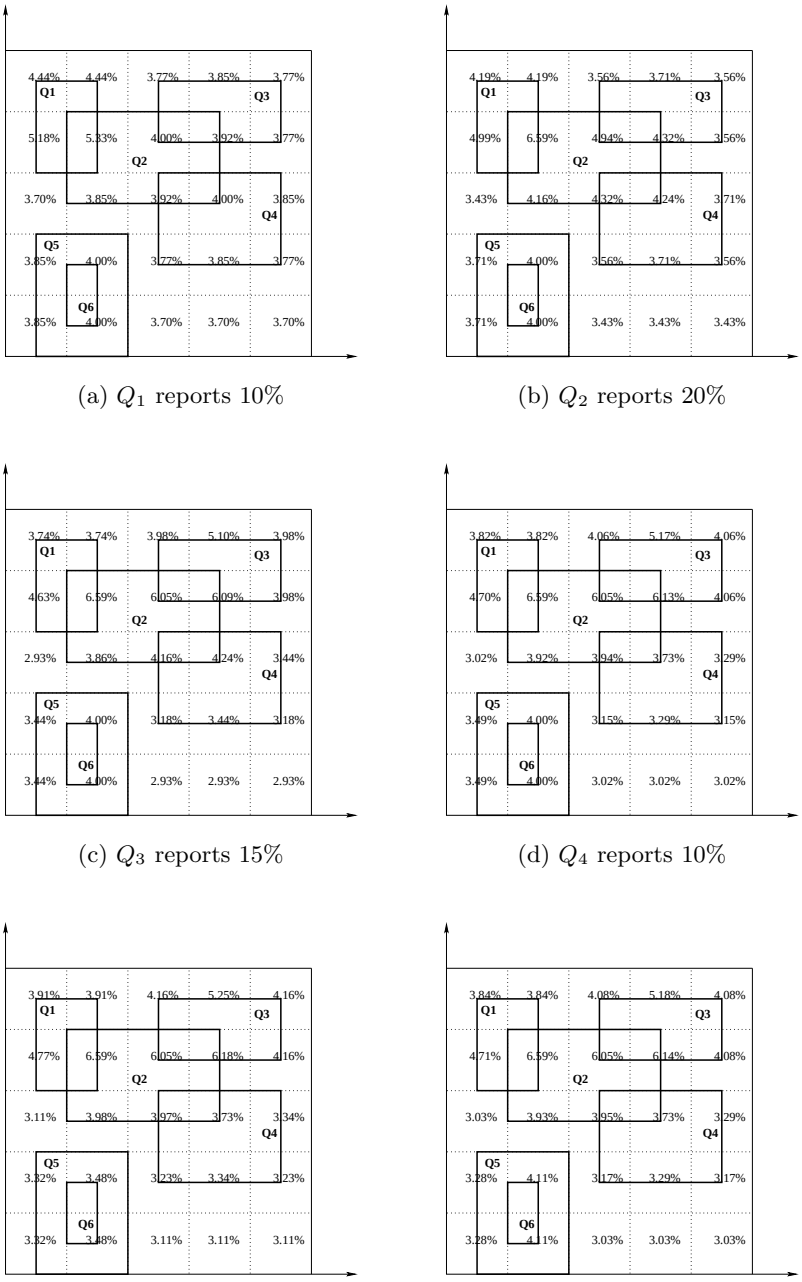
Fig. 5. Example of six continuous queries with an initial histogram

grid buckets. Each of these queries returns the vehicles of type "Truck" in its covering region (Figure 1). $Q2$ is overlapped by $Q1, Q3$, and $Q4$. $Q6$ is contained in $Q5$. Each grid bucket starts with a selectivity estimate of 4% (Figure 5(b)). Consider when the histogram manager receives feedback from the query executor that Q_1 reports its selectivity as 10%.

The selectivity estimate of Q_1 is 6% according to Equation 2. The difference 10-6 = 4% is distributed among those grid cells overlapped by Q_1. Consider the upper left grid cell C_{ul}. The normalized rate of Q_1 for C_{ul}'s is 0.25*0.04/0.06 = 0.1667. Thus $\mathcal{H}[1,1]$ = 0.04 + 0.1667*0.04 = 0.0467 = 4.67%. We still need to modify the histogram in order to reach the unity invariance. The increase (or decrease) of the selectivity estimate in a lit region should be accompanied with the decrease (or increase) of the selectivity estimate in the dark region. So, we decrease the selectivity estimate of the dark regions uniformly as much as the increase in the lit regions (4%). For instance, the lower-right bucket, C_{lr}, consists of 1/13.25 of the dark area. The current normalized rate of the **dark** region for C_{lr} is 0.0755. The new value for $\mathcal{H}[5,5]$ will be 0.04 - 0.0755*0.04 = 0.0370 = 3.70%. Figure 6(a) gives the histogram after refinement. The upper-left bucket has also a dark portion that results in decreasing $\mathcal{H}[1,1]$.

Figures 6(b) and 6(f) give the successive updates to the histogram due to the subsequent feedbacks that the histogram manager receives as follows: Q_2 reports 20%, Q_3 reports 15%, Q_4 reports 10%, Q_5 reports 10%, and Q_6 reports 3%.

As a validity check, note that after refining the histogram, we just get a better selectivity estimate for the query. The new estimate is not the same as $\mathcal{S}$. Also, better selectivity estimates are obtained for the dark regions. With the succession of the feedbacks that report (almost) the same selectivity, the estimate for the query converges to the feedback.



(a) Q_1 reports 10%

(b) Q_2 reports 20%

(c) Q_3 reports 15%

(d) Q_4 reports 10%

(e) Q_5 reports 10%

(f) Q_6 reports 3%

Fig. 6. Successive refinements of the ST-Histogram after receiving the feedback.

6 Query Optimization Issues

Inspired by the fact that time repeats itself, we use online periodicity detection to detect periodic patterns in the distribution of the moving objects over the time. This helps in enhancing the selectivity estimation in ST-Histograms. In fact, this is an instance of where data mining can fit evenly in query optimization.

6.1 Online Periodicity Mining

Periodicity mining is defined as the process of discovering frequent periodic patterns in an attempt towards predicting the future behavior in time series data [13]. In our context, a time series for each region in the space is formed by collecting the selectivity values over time, whether they are exact values collected from the queries, or estimates computed by the previous technique. If the selectivity values are quantized into nominal levels, and each level (e.g., high, medium, low) is denoted by a symbol (e.g., a, b, c), then the time series can be considered a sequence over a finite alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \cdots\}$.

Periodically, the histogram manager tries to see if any periodic pattern exists in any cell of the histogram. We use the periodicity mining technique in [13]. When we detect a periodic pattern in a dark cell of the grid, we no longer compute its selectivity estimate as being uniformly distributed on all the dark regions. Indeed, we call such cell "a shaded cell". Shaded cells are treated as if there is a query that covers the whole cell. Such query reports the selectivity of the cell according to the periodic pattern of its selectivity.

Note that the periodicity behavior of a region is considered only if this region does not fall inside any query. In other words, the intense of the periodicity light is too little to affect the total light intensity of an already lighted region, yet is enough to shade a dark region.

6.2 Dynamic Query Optimization

The equipment of the DBMS with ST-Histograms enables the existence of an adaptive query processor. With the online update of the ST-Histograms, we are able to detect when the currently executed query plan is suboptimal. In this case, a *need to re-optimize* flag is raised and the optimizer is reinvoked to compute a new optimal query execution plan to continue with. Hence, the already existing queries tune their pipeline for the current workload. Moreover, the new queries get benefit of the enhanced selectivity estimations (versus having one static histogram).

7 Experimental Results

We perform experiments to illustrate the efficiency of predicting the selectivity estimation using ST-Histograms. We use the Network-based Generator of Moving Objects [9] to generate a set of moving objects. The input to the generator is

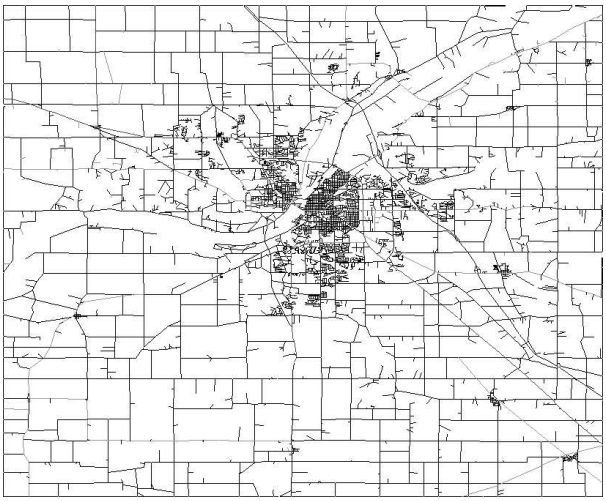


Fig. 7. Road network map of the Greater Lafayette Area City

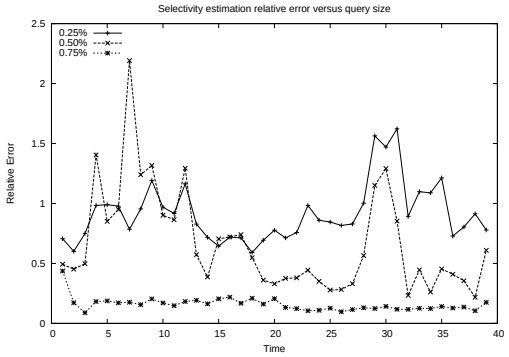
the road map of the Greater Lafayette Area (Home of Purdue University) given in Figure 7. The output of the generator is a set of moving points that move on the road network of the given city. Moving objects can be cars, cyclists, pedestrians, etc. We generate 5K moving objects and up to 80 continuous queries over 10x10 grid. Each moving object or query reports its new information (if changed) every 10 seconds.

7.1 Effect of Query Size on The Prediction

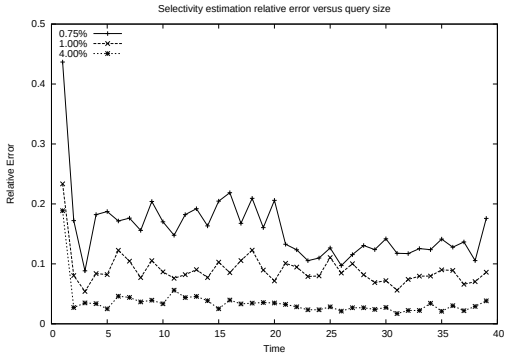
We measure the accuracy of the selectivity estimation of the existing queries by monitoring the relative error in estimating their selectivities. Let S_i and E_i be the actual and estimated selectivity of the ith query ($1 \leq i \leq M$), respectively, where M is the number of the queries. Equation 6 gives the relative error of estimating the selectivities of the existing queries.

$$\alpha = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\frac{S_i - E_i}{S_i}\right)^2} \tag{6}$$

Figures 8(a) and 8(b) give the performance of estimating the selectivity of the existing queries. α measures how accurate the prediction of the selectivity of the existing queries. From this experiment, we notice that smaller queries suffer from less accuracy than moderate to larger queries. When the queries are too small, many moving objects enter and exit the query range with high frequency. The high frequency of moving in and out the query range results in larger relative error. For a query of size 0.25% of the whole area, the average relative error is



(a) Small sized queries



(b) Moderate sized queries

Fig. 8. Performance of estimating the selectivity of the existing queries.

91.1% whereas for a query size of 0.5% the average relative error is 67.4%. This error is due to the existence of big dark portions in the grid cells corresponding to the queries. The smaller the dark portion, the less the relative error.

Moderate sized queries do not suffer from the fast moving objects. For moderate sized queries (1% of the whole range), the histogram manager is able to estimate the selectivity of the existing queries with an average relative error of 8.5%. The larger the query size the more accurate the estimation. Queries of size 4% give a relative error of 3.1%.

7.2 Coverage and The Prediction

A new query may come to the system in any area, whether lit or dark. We measure the accuracy of the prediction process for the selectivity of the new

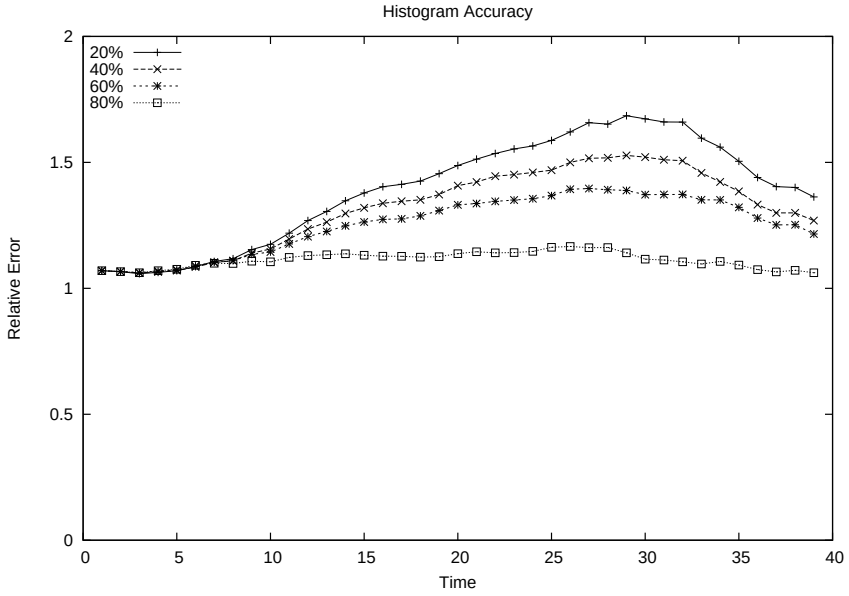


Fig. 9. Performance of estimating the selectivity of the new arriving queries

queries using the accuracy of the whole ST-Histogram. Equation 7 gives the average of the ratio between the estimated and the actual selectivities of all the histogram buckets, where S_{ij} is the actual selectivity of the grid cell $\mathcal{G}[i,j]$.

$$\beta = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\mathcal{H}[i,j]}{S_{ij}} \tag{7}$$

The selectivity estimate of existing queries is calculated by consulting lit buckets in the ST-Histogram. However, to get a selectivity estimate for a new arriving queries, dark buckets in the ST-Histogram may be consulted, yielding to higher relative error. Figure 9 gives the performance of estimating the selectivity of a bucket in an ST-Histogram (averaged over all buckets). β measures how accurate the selectivity estimates of any new arriving query. The histogram manager is able to estimate the current selectivity for any new query with an average error of 39% when the coverage are is 20%. In this experiment, the value of the error is due to a large number of grid cells totally dark (about 80% of the grid cells). The selectivity estimate of the dark region is uniformly distributed among those dark grid cells. The more spread the queries are on the space, the less number of complete dark cells, the less the relative error. When the coverage is 40%, the error is 31%. For coverage of 60%, the error is 25%, whereas the error is 11% for 80% coverage.

8 Conclusion

In this paper, we explored the usage of spatio-temporal histograms for selectivity estimation of spatio-temporal operators. We presented a general framework for building and continuously maintaining spatio-temporal histograms. The main idea of our proposed spatio-temporal histograms is to use a *continuous feedback* from the outstanding continuous queries to maintain a spatio-temporal histogram for only those parts of the spatial space that are of interest to at least one outstanding continuous query. Parts of the spatial space that are not of interest to any of the outstanding queries do not participate in maintaining the spatio-temporal histograms, thus the overhead of continuously maintaining our spatio-temporal histograms is reduced. Our proposed spatio-temporal histograms utilize periodicity detection techniques to discover temporal periodic patterns. Discovering temporal patterns provides pre-computation of the optimal query plan over the course of execution of continuous queries. Experimental results show that our spatio-temporal histograms provide only 8.5% error for the existing queries of size 1%. An average error of 25% for new queries when the existing query coverage is 60%.

References

1. FCC: Enhanced 911 - Wireless Services. http://www.fcc.gov/911/enhanced/.
2. A. Aboulnaga and S. Chaudhuri. Self-tuning Histograms: Building Histograms without Looking at Data. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 181–192. ACM Press, 1999.
3. N. An, Z.-Y. Yang, and A. Sivasubramaniam. Selectivity Estimation for Spatial Joins. In *ICDE '01: Proceedings of the 17th International Conference on Data Engineering*, pages 368–375. IEEE Computer Society, 2001.
4. W. G. Aref, S. E. Hambrusch, and S. Prabhakar. Pervasive Location Aware Computing Environments (PLACE). http://www.cs.purdue.edu/place/, 2003.
5. W. G. Aref and H. Samet. Estimating Selectivity Factors of Spatial Operations. In *Proceedings, Optimization in Databases - 5th Int'l Workshop on Foundations of Models and Languages for Data and Objects*, pages 31–43, Sep 1993.
6. W. G. Aref and H. Samet. A Cost Model for Query Optimization using R-trees. In *ACM-GIS '94: Proceedings of the ACM Workshop on Advances in Geographic Information Systems*, Dec 1994.
7. A. Belussi, E. Bertino, and A. Nucita. Grid based methods for estimating spatial join selectivity. In *GIS '04: Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pages 92–100. ACM Press, 2004.
8. A. Belussi and C. Faloutsos. Estimating the Selectivity of Spatial Queries Using the 'Correlation' Fractal Dimension. In *VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases*, pages 299–310. Morgan Kaufmann Publishers Inc., 1995.
9. T. Brinkhoff. A Framework for Generating Network-Based Moving Objects. *Geoinformatica*, 6(2):153–180, 2002.
10. F. Buccafurri and G. Lax. Fast range query estimation by N-level tree histograms. *Data Knowl. Eng.*, 51(2):257–275, 2004.

11. C. M. Chen and N. Roussopoulos. Adaptive Selectivity Estimation using Query Feedback. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, pages 161–172. ACM Press, 1994.
12. Y.-J. Choi and C.-W. Chung. Selectivity Estimation for Spatio-temporal Queries to Moving Objects. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 440–451. ACM Press, 2002.
13. M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid. Incremental, Online, and Merge Mining of Partial Periodic Patterns in Time-Series Databases. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(3):332–342, 2004.
14. C. Faloutsos, B. Seeger, A. Traina, and J. Caetano Traina. Spatial Join Selectivity using Power Laws. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 177–188. ACM Press, 2000.
15. P. B. Gibbons, Y. Matias, and V. Poosala. Fast Incremental Maintenance of Approximate Histograms. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 466–475. Morgan Kaufmann Publishers Inc., 1997.
16. G. Graefe and W. J. McKenna. The Volcano Optimizer Generator: Extensibility and Efficient Search. In *ICDE '93: Proceedings of the Ninth International Conference on Data Engineering*, pages 209–218. IEEE Computer Society, 1993.
17. P. J. Haas and A. N. Swami. Sequential Sampling Procedures for Query Size Estimation. In *SIGMOD '92: Proceedings of the 1992 ACM SIGMOD international conference on Management of data*, pages 341–350. ACM Press, 1992.
18. P. J. Haas and A. N. Swami. Sampling-Based Selectivity Estimation for Joins Using Augmented Frequent Value Statistics. In *ICDE '95: Proceedings of the Eleventh International Conference on Data Engineering*, pages 522–531. IEEE Computer Society, 1995.
19. M. Hadjieleftheriou, G. Kollios, D. Gunopulos, and V. J. Tsotras. On-Line Discovery of Dense Areas in Spatio-temporal Databases. In *SSTD '03: Proceedings of the 8th International Symposium on Spatial and Temporal Databases*, pages 306–324, Jul 2003.
20. M. Hadjieleftheriou, G. Kollios, and V. J. Tsotras. Performance Evaluation of Spatio-temporal Selectivity Estimation Techniques. In *SSDBM '03: Proceedings of the 15th International Conference on Scientific and Statistical Database Management*, pages 202–211, 2003.
21. I. F. Ilyas, H. G. Elmongui, and W. G. Aref. Adaptive Processing of Ranking Queries. Technical Report CSD-TR-05-002, Purdue University, 2005.
22. G. S. Iwerks, H. Samet, and K. Smith. Continuous K-Nearest Neighbor Queries for Continuously Moving Points with Updates. In *VLDB '03: Proceedings of the 29th International Conference on Very Large Data Bases*, Sep 2003.
23. N. Kabra and D. J. DeWitt. Efficient Mid-query Re-optimization of Sub-optimal Query execution Plans. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 106–117. ACM Press, 1998.
24. R. P. Kooi. *The Optimization of Queries in Relational Databases*. PhD thesis, Case Western Reserve University, 1980.
25. D. Kwon, S. Lee, and S. Lee. Indexing the Current Positions of Moving Objects Using the Lazy Update R-tree. In *MDM '02: Proceedings of the Third International Workshop on Multimedia Data Mining*, pages 113–120, Jan 2002.
26. I. Lazaridis, K. Porkaew, and S. Mehrotra. Dynamic Queries over Mobile Objects. In *EDBT '02: Proceedings of the 8th International Conference on Extending Database Technology*, pages 269–286, Mar 2002.

27. M.-L. Lee, W. Hsu, C. S. Jensen, and K. L. Teo. Supporting Frequent Updates in R-Trees: A Bottom-Up Approach. In *VLDB '03: Proceedings of the 29th International Conference on Very Large Data Bases*, Sep 2003.
28. R. J. Lipton and J. F. Naughton. Query Size Estimation by Adaptive Sampling. *Journal of Computer and System Sciences*, 51(1):18–25, 1995.
29. R. J. Lipton, J. F. Naughton, and D. A. Schneider. Practical Selectivity Estimation through Adaptive Sampling. In *SIGMOD '90: Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, pages 1–11. ACM Press, 1990.
30. N. Mamoulis and D. Papadias. Selectivity Estimation of Complex Spatial Queries. In *SSTD '01: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, pages 155–174. Springer-Verlag, 2001.
31. G. S. Manku, S. Rajagopalan, and B. G. Lindsay. Approximate Medians and other Quantiles in One Pass and with Limited Memory. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 426–435. ACM Press, 1998.
32. Y. Matias, J. S. Vitter, and M. Wang. Wavelet-based histograms for selectivity estimation. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 448–459. ACM Press, 1998.
33. M. F. Mokbel and W. G. Aref. GPAC: Generic and Progressive Processing of Mobile Queries over Mobile Data. In *MDM '05: Proceedings of the Third International Workshop on Multimedia Data Mining*, May 2005.
34. M. F. Mokbel, W. G. Aref, S. E. Hambrusch, and S. Prabhakar. Towards Scalable Location-aware Services: Requirements and Research Issues. In *ACM-GIS '03: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems*, pages 110–117, Nov 2003.
35. M. F. Mokbel, X. Xiong, W. G. Aref, S. Hambrusch, S. Prabhakar, and M. Hammad. PLACE: A Query Processor for Handling Real-time Spatio-temporal Data Streams (Demo). In *VLDB '04: Proceedings of the Thirtieth International Conference on Very Large Data Bases*, Aug 2004.
36. M. Muralikrishna and D. J. DeWitt. Equi-depth Multidimensional Histograms. In *SIGMOD '88: Proceedings of the 1988 ACM SIGMOD international conference on Management of data*, pages 28–36. ACM Press, 1988.
37. T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode. TrafficView: A Scalable Traffic Monitoring System. In *MDM '04: Proceedings of the 5th IEEE International Conference on Mobile Data Management*, Jan 2004.
38. G. Piatetsky-Shapiro and C. Connell. Accurate Estimation of the Number of Tuples Satisfying a Condition. In *SIGMOD '84: Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 256–276. ACM Press, 1984.
39. V. Poosala. *Histogram-based Estimation Techniques in Database Systems*. PhD thesis, University of Wisconsin at Madison, 1997.
40. V. Poosala, P. J. Haas, Y. E. Ioannidis, and E. J. Shekita. Improved Histograms for Selectivity Estimation of Range Predicates. In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 294–305. ACM Press, 1996.
41. V. Poosala and Y. E. Ioannidis. Selectivity Estimation Without the Attribute Value Independence Assumption. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 486–495. Morgan Kaufmann Publishers Inc., 1997.

42. S. Prabhakar, Y. Xia, D. V. Kalashnikov, W. G. Aref, and S. E. Hambrusch. Query Indexing and Velocity Constrained Indexing: Scalable Techniques for Continuous Queries on Moving Objects. *IEEE Transactions on Computers*, 51(10):1124–1140, Oct 2002.
43. G. Proietti and C. Faloutsos. Selectivity Estimation of Window Queries for Line Segment Datasets. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, pages 340–347. ACM Press, 1998.
44. S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the Positions of Continuously Moving Objects. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 331–342, May 2000.
45. P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access Path Selection in a Relational Database Management System. In *SIGMOD '79: Proceedings of the 1979 ACM SIGMOD international conference on Management of data*, pages 23–34. ACM Press, 1979.
46. C. Sun, D. Agrawal, and A. E. Abbadi. Selectivity Estimation for Spatial Joins with Geometric Selections. In *EDBT '02: Proceedings of the 8th International Conference on Extending Database Technology*, pages 609–626. Springer-Verlag, 2002.
47. Y. Tao, D. Papadias, and Q. Shen. Continuous Nearest Neighbor Search. In *VLDB '02: Proceedings of the 28th International Conference on Very Large Data Bases*, pages 287–298, Aug 2002.
48. Y. Tao, D. Papadias, and J. Sun. The TPR*-Tree: An Optimized Spatio-temporal Access Method for Predictive Queries. In *VLDB '03: Proceedings of the 29th International Conference on Very Large Data Bases*, Sep 2003.
49. Y. Tao, D. Papadias, J. Zhai, and Q. Li. Venn Sampling: A Novel Prediction Technique for Moving Objects. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering*, April 2005.
50. Q. Zhang and X. Lin. Clustering Moving Objects for Spatio-temporal Selectivity Estimation. In *CRPIT '27: Proceedings of the fifteenth conference on Australasian database*, pages 123–130. Australian Computer Society, Inc., 2004.