

# Estimating the Result Size of a Query to Velocity Skewed Moving Objects

Yong-Jin Choi<sup>a</sup>      Ho-Hyun Park<sup>b</sup>      Chin-Wan Chung<sup>a</sup>

<sup>a</sup>*Division of Computer Science  
Department of Electrical Engineering and Computer Science  
Korea Advanced Institute of Science and Technology  
373-1, Kusong-dong, Yusong-gu, Taejeon 305-701, South Korea  
{omni, chungcw}@islabs.kaist.ac.kr*

<sup>b</sup>*School of Electrical and Electronics Engineering  
Chung-Ang University  
221, HukSuk-Dong, DongJae-Gu, Seoul 156-756, South Korea  
hohyun@cau.ac.kr*

*Key words:* Spatio-Temporal Databases, Moving Object, Selectivity Estimation

---

## 1 Introduction

With new developments in positioning systems and electronics, various researches for moving objects have progressed [3,5,6]. Our paper is related to selection queries considering future positions of moving objects, which are referred to as future queries [5]. An example of the future query is as follows: “which airplanes will be inside a query window 20 minutes from now?” Airplanes correspond to moving objects that move as time passes. Recently, a data model that can deal with the future locations of an object has been proposed [5]. Like various studies [2,4,6], our work is based on this model.

In order to process the selection query efficiently, an accurate estimation of the selectivity is required. The selectivity is defined as the ratio of the number of data in the query result to the total number of data in the database. The query optimizer chooses the most efficient execution plan among all possible plans by estimating the cost of each plan. The accuracy of the selectivity estimation significantly affects the selection of an efficient plan. Recently, an effective method of selectivity estimation for the future locations of moving objects has been proposed [2]. The estimation method used the spatio-temporal histogram for the selectivity estimation. The construction algorithm of the histogram was focused on the spatial locations of moving objects, in contrast to both their locations and velocities. The approach may not

be very effective about moving objects with very skewed velocity distributions. In this paper, we construct the spatio-temporal histogram to balance the two aspects of moving objects.

Experimental results show that our proposed spatio-temporal histogram method has better estimation results over various queries for synthetic moving objects. In experiments on moving objects with a very skewed velocity distribution, while the existing method for the selectivity estimation of a spatio-temporal query has the average error ratio from 18% to 59%, our method has the average error ratio from 12% to 23%.

## 2 Related Work

We briefly describe the selectivity estimation considering the future locations of moving objects [2]. The method used the spatio-temporal histogram to estimate the number of objects passing the query. The histogram method is one of the most popular selectivity estimation methods, because it approximates any data distribution and requires reasonably small storage with low error rates [1]. The spatio-temporal histogram is a set of spatio-temporal buckets. The spatio-temporal bucket consists of a spatial bounding rectangle, a velocity bounding rectangle, and the number of objects in two bounding rectangles. Table 1 presents the symbols used throughout the paper. Subscripts 1 and 2 of  $SB$ ,  $VB$ , and  $QB$  represent dimensions in the 2D space. Figure 1(a) shows a spatio-temporal query. In Figure 1(b), dotted thick lines indicate a spatio-temporal bucket. Like the uniformity assumption in each bucket of the histogram [1], a general uniform distribution in each spatio-temporal bucket of the spatio-temporal histogram is assumed. That is, the spatial locations of moving objects are distributed uniformly in the spatial bounding rectangle and the velocities of moving objects are distributed uniformly in the velocity bounding rectangle. According to this assumption, analytical formulas for the selectivity estimation for a spatio-temporal query intersecting spatio-temporal buckets has been proposed [2]. Moving objects are updated very frequently. If new update information is not reflected to the histogram, the query optimizer may choose an inefficient plan because an inaccurate estimation of selectivity is used. The histogram is not required to be updated from the whole objects, but it can be updated from sample objects. In spatio-temporal databases, the histogram can be maintained practically by using sample data [2].

The method in [2] creates the spatial bounding rectangles of a spatio-temporal histogram using the MinSkew algorithm [1] based on the locations of moving objects. The MinSkew algorithm creates the buckets of a histogram to minimize the spatial-skew of spatial objects. As shown in Figure 2, consider a spatio-temporal bucket of the spatio-temporal histogram. Dots indicate the spatial positions of moving objects. Arrows(or velocity vectors) indicate the future positions of moving objects. A moving object  $o_1$  has the velocity vector (8,1). The rectangle means  $SB$  of the spatio-temporal bucket. As shown in Figure 2, the spatial positions of objects can be uniformly distributed in  $SB$  by the MinSkew algorithm. The velocity bounding rectangle of a spatio-temporal bucket has minimum and maximum velocity values of moving objects in the spatial bounding rectangle of the bucket. So, as shown in

Table 1

Symbol description

Symbol	Description
$I(I^l, I^h)$	interval: $I^l \leq I^h$ ; $I^l$ , low value of $I$ ; $I^h$ , high value of $I$
$t^u$	histogram update time
$QB(a_1, a_2)$	spatial rectangle of query: $a_1, a_2$ are intervals
$Q(QB, t)$	query: time interval $t$
$SB(s_1, s_2)$	spatial bounding rectangle: $s_1, s_2$ are intervals
$VB(v_1, v_2)$	velocity bounding rectangle: $v_1, v_2$ are intervals
$B_{st}(n_{st}, SB, VB)$	spatio-temporal bucket: $n_{st}$ , number of moving objects in $SB$ and $VB$
$B_s(n_s, SB)$	spatial bucket: $n_s$ , number of moving objects in $SB$
$H_s$	spatial histogram: a set of $B_s$
$B_v(n_v, VB)$	velocity bucket: $n_v$ , number of moving objects in $VB$
$H_v$	velocity histogram: a set of $B_v$
$H_{st}(H_s, H_v)$	spatio-temporal histogram
$N_B$	number of $B_s$ in $H_s$ (or $B_v$ in $H_v$ )
$N$	number of moving objects
$N_{sm}$	number of sample moving objects

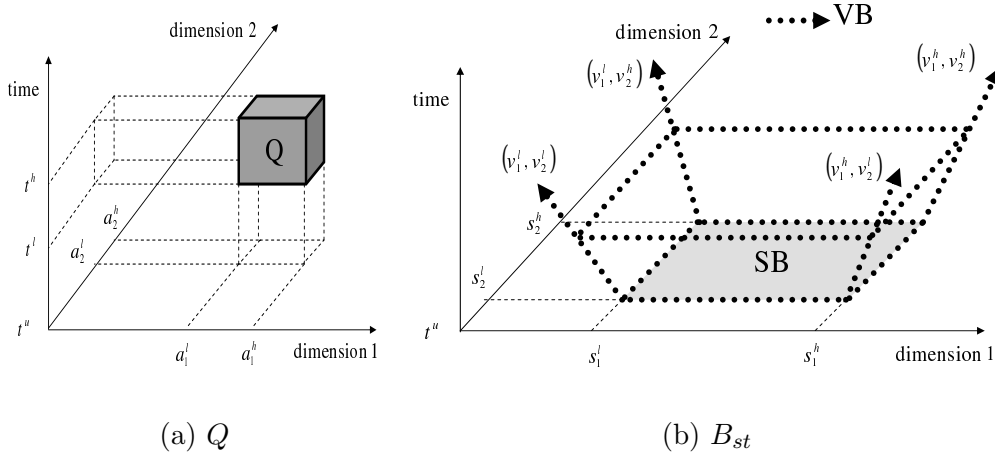


Fig. 1. Spatio-temporal query and spatio-temporal bucket

Figure 2,  $v_1$  and  $v_2$  of  $VB$  are  $[-6, 8]$  and  $[-7, 2]$ , respectively. Although the velocity distribution of moving objects is actually skewed in  $VB$ , the selectivity is estimated under the uniformity assumption in  $VB$ . As a result, the approach in [2] may not be very effective for moving objects with a very skewed velocity distribution.

### 3 Improved Spatio-Temporal Histogram for Selectivity Estimation

In this section, we describe how to construct our improved spatio-temporal histogram. Our improved spatio-temporal histogram  $H_{st}$  consists of a spatial histogram  $H_s$  and a velocity

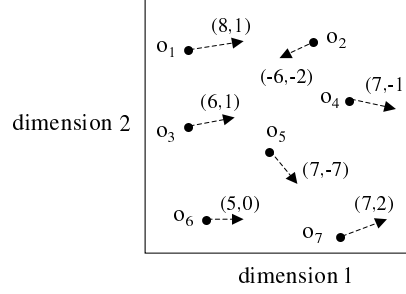


Fig. 2. A spatio-temporal bucket containing moving objects with a very skewed velocity distribution

histogram  $H_v$ . As shown in Figure 3, we propose an efficient construction algorithm of the spatio-temporal histogram considering both locations and velocities of moving objects. We create a spatial histogram  $H_s$  using the MinSkew algorithm with a spatial grid  $SG$ . A grid, a set of cells, can be represented by an array structure. Figure 4 illustrates how to create a spatial histogram with 3 buckets. Figure 4(a) shows the spatial locations of 25 objects. Figure 4(b) shows a spatial grid  $SG$  after processing lines 1-3 of Figure 3. Let's assume that Figure 4(c) is the result after processing line 6. Then, Figure 4(d) shows the result after processing lines 8-9. Similarly, we also create a velocity histogram  $H_v$  using the MinSkew algorithm with a velocity grid  $VG$ . So, two bounding rectangles of our spatio-temporal bucket conceptually consist of an  $SB$  of  $H_s$  and a  $VB$  of  $H_v$ . We can use  $H_v$  with the minimized velocity-skew because  $H_v$  is created by the MinSkew algorithm.

**Algorithm** constructSTH( $t^u, SM, N_B$ )

**Input:**  $t^u$ , histogram update time;  $SM$ , a set of sample moving objects;

$N_B$ , number of  $B_s$  in  $H_s$  (or  $B_v$  in  $H_v$ )

**Output:**  $H_{st}(H_s, H_v)$ , spatio-temporal histogram

01. for each moving object  $m$  in  $SM$ ,
02.   let  $s$  be the spatial position of  $m$  at  $t^u$ .
03.   increase the number of objects of a cell, in  $SG$ , which contains  $s$ .
04.   let  $v$  be the velocity point on 2D space of  $m$ .
05.   increase the number of objects of a cell, in  $VG$ , which contains  $v$ .
06. construct  $N_B$  spatial bounding rectangles of  $H_s$  from  $SG$  using the MinSkew algorithm.
07. construct  $N_B$  velocity bounding rectangles of  $H_v$  from  $VG$  using the MinSkew algorithm.
08. for each  $B_s(n_s, SB)$  of  $H_s$ ,
09.   assign the sum of numbers of objects of  $SG$ 's cells in  $SB$  to  $n_s$ .
10. for each  $B_v(n_v, VB)$  of  $H_v$ ,
11.   assign the sum of numbers of objects of  $VG$ 's cells in  $VB$  to  $n_v$ .

Fig. 3. Improved  $H_{st}$  construction algorithm

Next, we define a function  $CQ$  needed to process the selectivity estimation algorithm using  $H_{st}$ . Let  $t^u$  be the histogram update time,  $v$  the velocity bounding interval,  $a$  the spatial interval of the query, and  $t$  the time interval of the query.  $CQ$  calculates  $I$  which is the maximum spatial interval at  $t^u$  of moving objects that can pass the query. For example, consider a query and moving objects limited by a velocity bounding interval  $v[v^l < 0, v^h > 0]$ . As shown in Figure 5, all movements within the velocity interval  $v$  of an object that starts from a point  $(p, t^u)$  cannot pass the query. From  $CQ$ , the thick line  $I$  depicts the maximum spatial interval of moving objects that can pass the query.

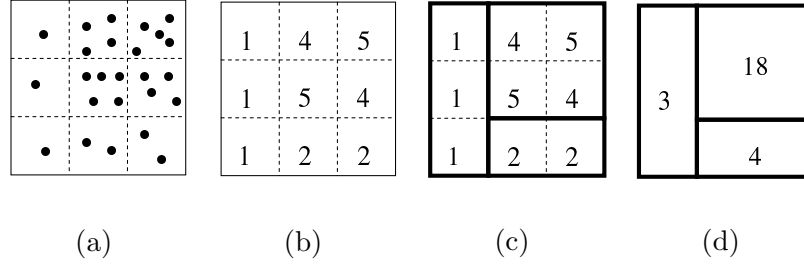


Fig. 4. Histogram construction

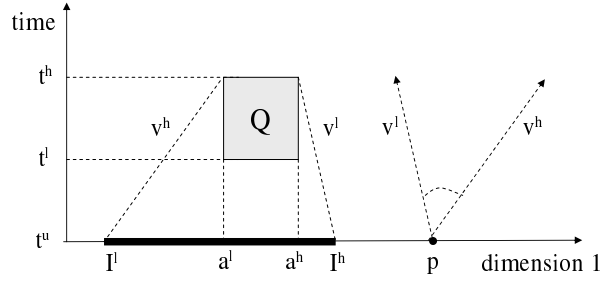


Fig. 5. Maximum spatial interval  $I$  derived by CQ

$CQ(t^u, v, a, t)$  produces an interval  $I$  as follows:

$$I^l = \begin{cases} a^l - (t^l - t^u)v^h & \text{if } v^h < 0 \\ a^l - (t^h - t^u)v^h & \text{otherwise} \end{cases} \quad I^h = \begin{cases} a^h - (t^l - t^u)v^l & \text{if } v^l > 0 \\ a^h - (t^h - t^u)v^l & \text{otherwise} \end{cases}$$

Next, we explain how to estimate a selectivity for a spatio-temporal query intersecting a spatio-temporal bucket with  $n$  moving objects bounded by  $SB$  and  $VB$  as shown in Figure 6. A formula  $(t^h - t^u)(v_1^h - v_1^l)(t^h - t^u)(v_2^h - v_2^l)$  indicates an area of all the possible positions at  $t^h$  of a moving object that starts from a point in  $SB$ . Then  $(s_1^h - s_1^l)(t^h - t^u)(v_1^h - v_1^l)(s_2^h - s_2^l)(t^h - t^u)(v_2^h - v_2^l)$  corresponds to all the movements of all the moving objects that start from  $SB$ . To consider all the movements of moving objects that pass the query, we present a function  $\mathbf{A}$  that is an area of a sub-rectangle of  $SB$  where moving objects that start from the sub-rectangle pass one point in five sides of the query. For five sides of the query,  $\mathbf{A}$  has different formulas: **A1**, **A2**, **A3**, **A4**, and **AB**.  $\int_{t^l}^{t^h} \int_{a_2^l}^{a_2^h} \mathbf{A2}(\lambda_1, \lambda_2) d\lambda_1 d\lambda_2$  corresponds to all the movements of moving objects that start from a sub-rectangle of  $SB$  and first pass the  $a_1^h$  side of the query, as illustrated in Figure 6. Similarly,  $\int_{t^l}^{t^h} \int_{a_2^l}^{a_2^h} \mathbf{A1}(\lambda_1, \lambda_2) d\lambda_1 d\lambda_2$ ,  $\int_{t^l}^{t^h} \int_{a_1^l}^{a_1^h} \mathbf{A3}(\lambda_1, \lambda_2) d\lambda_1 d\lambda_2$ ,  $\int_{t^l}^{t^h} \int_{a_1^l}^{a_1^h} \mathbf{A4}(\lambda_1, \lambda_2) d\lambda_1 d\lambda_2$ , and  $\int_{a_2^l}^{a_2^h} \int_{a_1^l}^{a_1^h} \mathbf{AB}(\lambda_1, \lambda_2) d\lambda_1 d\lambda_2$  are for the  $a_1^l$  side, the  $a_2^l$  side, the  $a_2^h$  side, and the base side of the query, respectively. Let  $STBQ(t^u, n, SB, VB, Q)$  be a selectivity estimation function between a spatio-temporal bucket and a spatio-temporal query, where  $n$  is the number of moving objects bounded by  $SB$  and  $VB$ .  $STBQ(t^u, n, SB, VB, Q)$  is estimated as follows:

$$n \times \frac{\int_{t^l}^{t^h} \int_{a_2^l}^{a_2^h} (\mathbf{A1}(\lambda_1, \lambda_2) + \mathbf{A2}(\lambda_1, \lambda_2)) d\lambda_1 d\lambda_2 + \int_{t^l}^{t^h} \int_{a_1^l}^{a_1^h} (\mathbf{A3}(\lambda_1, \lambda_2) + \mathbf{A4}(\lambda_1, \lambda_2)) d\lambda_1 d\lambda_2 + \int_{a_1^l}^{a_1^h} \int_{a_2^l}^{a_2^h} \mathbf{AB}(\lambda_1, \lambda_2) d\lambda_1 d\lambda_2}{(s_1^h - s_1^l)(t^h - t^u)(v_1^h - v_1^l)(s_2^h - s_2^l)(t^h - t^u)(v_2^h - v_2^l)}$$

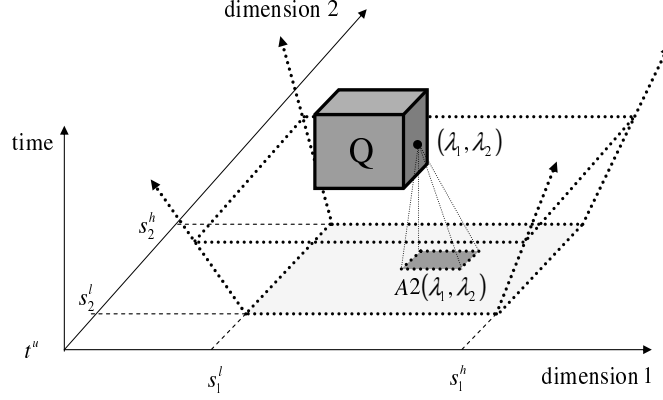


Fig. 6. **A2**

Figure 7 shows a selectivity estimation algorithm using our spatio-temporal histogram  $H_{st}(H_s, H_v)$ . The estimation is calculated for each  $B_s$  of  $H_s$ . Let  $V_1$  and  $V_2$  be velocity bounding intervals with the minimum and maximum values of whole moving objects for dimension 1 and dimension 2, respectively. Lines 3-5 check whether a spatio-temporal bucket bounded by  $SB(s_1, s_2)$  and  $VB(V_1, V_2)$  intersects with the query  $Q$ . Lines 7-8 summate the estimation results between a  $B_s$  and each  $B_v$  of  $H_v$ . In line 8, the number of moving objects bounded by an  $SB$  and a  $VB$  is  $n_s \times \frac{n_v}{N_{sm}}$ . In line 10 of Figure 7, the final result of selectivity estimation is obtained by multiplying the sample factor  $\frac{N}{N_{sm}}$ .

**Algorithm** selectivityST

**Input:**  $t^u$ , histogram update time;  $H_{st}(H_s, H_v)$ , spatio-temporal histogram;  $Q$ , query

**Output:**  $sum$ , the result of selectivity estimation

```

01.  $sum \leftarrow 0$ .

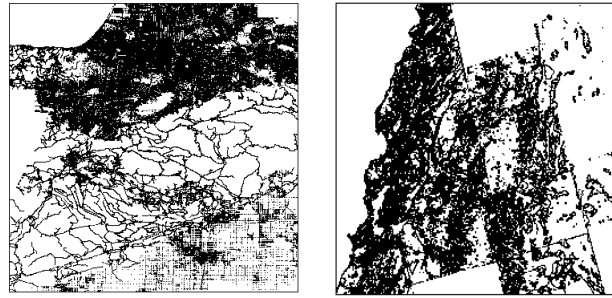
02. for each  $B_s(n_s, SB)$  in the  $H_s$ ,
03.    $I_1 \leftarrow s_1 \cap CQ(t^u, V_1, a_1, t)$ .
04.    $I_2 \leftarrow s_2 \cap CQ(t^u, V_2, a_2, t)$ .
05.   if  $I_1 \neq \emptyset \wedge I_2 \neq \emptyset$ ,
06.      $sum1 \leftarrow 0$ .
07.     for each  $B_v(n_v, VB)$  in the  $H_v$ ,
08.        $sum1 \leftarrow sum1 + STBQ(t^u, n_s \times \frac{n_v}{N_{sm}}, SB, VB, Q)$ .
09.      $sum \leftarrow sum + sum1$ .

10.  $sum \leftarrow sum \times \frac{N}{N_{sm}}$ .
```

Fig. 7. Selectivity estimation algorithm

## 4 Experiments

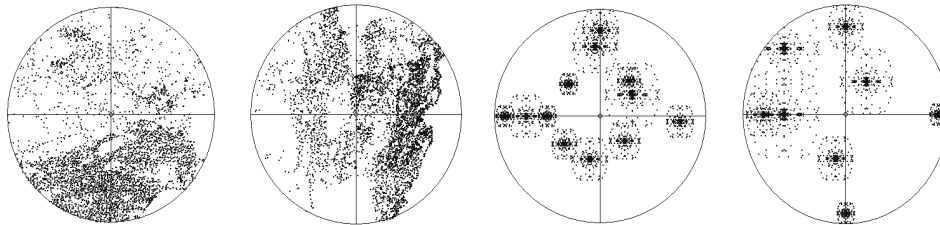
We make an experimental environment such that moving objects have skewed velocity distributions. Objects with the maximum speed 3.0 are set to move in the  $1000 \times 1000$  2D space. We use real-life spatial data to generate moving objects. Figure 8(a) and Figure 8(b) show the initial spatial locations of moving objects using Tiger/lines and Sequoia, respectively. Tiger/lines and Sequoia have been popularly used in spatial database research. The number of objects for Tiger/lines is 498,830. The number of objects for Sequoia is 547,401. Figure 9 depicts the velocity distributions of moving objects. For the velocity of a moving object, we randomly choose a point in a circle with a radius 3 corresponding to the maximum speed. The direction from the center point to the chosen point indicates the direction of the object. The distance between two points indicates the speed of the object. We use again Tiger/lines(Figure 9(a)) and Sequoia(Figure 9(b)) to generate the movements of objects with reasonably skewed velocity distributions. We also use synthetic data(Figure 9(c) and Figure 9(d)) to generate the movements of objects with very skewed velocity distributions. We experiment with 8 data sets of moving objects: All combinations for 2 types(S1,S2) of initial spatial locations and 4 types(V1,V2,V3,V4) of velocities.



(a) S1

(b) S2

Fig. 8. Initial spatial locations



(a) V1

(b) V2

(c) V3

(d) V4

Fig. 9. Velocities

We update approximately 1% of the whole moving objects every time unit and randomly select the updating data from the whole moving objects. We use the sample data from the whole moving objects using a simple round robin scheme for the spatio-temporal histogram

and update the spatio-temporal histogram from the sample data every time unit. This histogram update strategy can be possible because the I/O overhead for the spatio-temporal histogram update from the sample data is remarkably low, compared to the update overhead of approximately 1% of the whole moving objects per time unit. The size of  $H_{st}$  is set to 4 Kbytes, which corresponds to a page size. A spatial grid ( $SG$ ) for constructing the spatio-temporal histogram is set to  $50 \times 50$ . The size of the query spatial rectangle ( $QB$ ) varies as 0.1%, 0.25%, 0.5%, 1% of the size of the data space. The spatial position of a query is randomly chosen in the data space.

Figure 10 shows the experimental results for moving objects with reasonably skewed velocity distributions. Let ISTH be our Improved Spatio-Temporal Histogram and STH be the Spatio-Temporal Histogram [2]. Figure 10(a) and Figure 10(a) show the experimental results for (S1,V2) and (S2,V1), respectively. Each query is evaluated by the average relative error of 200 queries. The time interval length is set to 15. In general, our proposed method more accurately estimates the selectivity of spatio-temporal queries to moving objects, compared to the existing method.

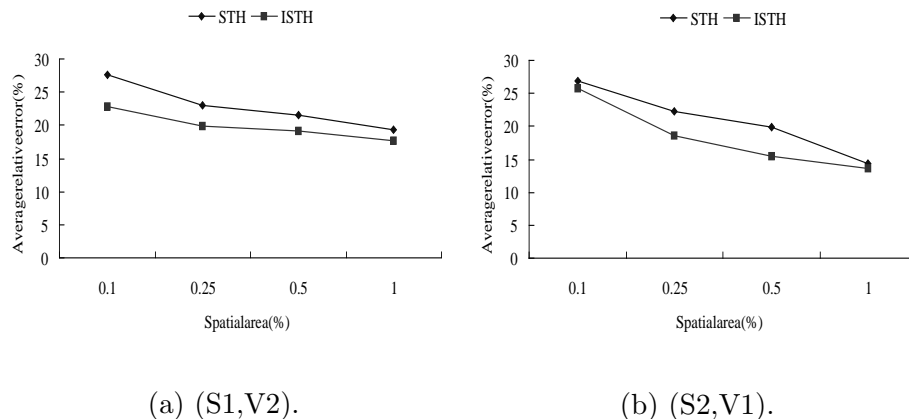


Fig. 10. Average relative error for ISTH and STH to moving objects with reasonably skewed velocity distributions

We also consider experimental results for moving objects with very skewed velocity distributions. Figure 11(a) and Figure 11(a) show experimental results for (S1,V3) and (S2,V4), respectively. The time interval length is set to 30. As shown in Figure 11, our proposed method has considerably better estimation results, compared to the existing method.

## 5 Conclusions

Spatio-temporal databases have been studied intensively in recent years. In this paper, we proposed the construction algorithm of an improved spatio-temporal histogram to accurately estimate the selectivity for spatio-temporal queries for the future locations of moving objects. Our algorithm constructs a spatio-temporal histogram to balance spatial location and

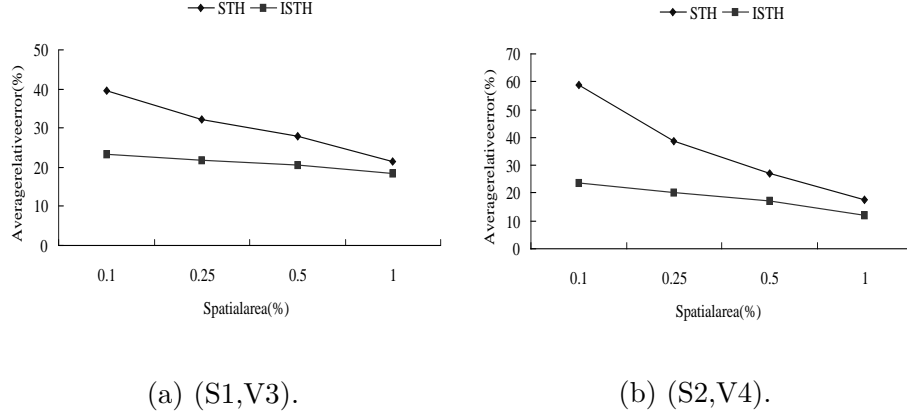


Fig. 11. Average relative error for ISTH and STH to moving objects with very skewed velocity distributions

velocity distributions of moving objects. In experiments on moving objects with very skewed velocity distributions, our proposed method produced a significantly accurate estimation of the selectivity for spatio-temporal queries, compared to the existing method.

### Acknowledgement

This work was supported by University Research Program of Ministry of Information & Communication in Republic of Korea.

### References

- [1] S. Acharya, V. Poosala, and S. Ramaswamy. Selectivity estimation in spatial databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 13–24, 1999.
- [2] Y. J. Choi and C. W. Chung. Selectivity estimation for spatio-temporal queries to moving objects. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 440–451, 2002.
- [3] D. Pfoser, C. S. Jensen, and Y. Theodoridis. Novel approaches in query processing for moving object trajectories. In *Proceedings of the International Conference on Very Large Data Bases*, pages 395–406, 2000.
- [4] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the positions of continuously moving objects. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 331–342, 2000.
- [5] A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Modeling and querying moving objects. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 422–432, 1997.
- [6] Y. Tao and D. Papadias. Time-parameterized queries in spatio-temporal databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 334–345, 2002.