EFFECTIVE LOCATION ACQUISITION CONTROL ALGORITHMS FOR THE LOCATION-BASED ALERT SERVICES IN MOBILE ENVIRONMENTS

So-Young Kang[1], Kwang-Jo Lee[1], Sung-Bong Yang[1]

[1]Department of Computer Science, Yonsei Univ. Seoul, 126-749 Korea
{milkyway, kjlee5435, yang}@cs.yonsei.ac.kr

ABSTRACT

The location-based alert services can be regarded as the one of the most practical location-based services. For the services, an alert service system for the services alerts mobile device users when they enter into or leave from predefined specific regions, and provides certain services previously asked by the users for special purposes such as security. For providing proper services the alert service system should acquire the location information of the users periodically. However, the system that handles the locations of the users may face serious problems as the number of users increases fast. Hence it is a critical issue to properly adjust the time interval of location data acquisitions while maintaining the accuracy of the services. In this paper we propose effective location acquisition algorithms; the speed-based acquisition algorithm, the angle–based acquisition algorithm, and the hybrid algorithm combining the speed with the angle-based algorithms. We also present three grid-based acquisition algorithms in which a longer time interval is used when a user is not near the alert areas. The proposed algorithms could reduce the amount of location information to be acquired based on the movement of the users. The average numbers of location acquisitions of the speed-based, the angle–based, and the hybrid algorithms were reduced by 19.2%, 35.8%, and 35.6% over the distance-based algorithm, respectively, while they maintained the almost same level of accuracy. Among the grid-based algorithms, the grid-angle acquisition algorithm further improved the average number of acquisitions by 5.2% over the angle-based algorithm, which is 41.0% improvement over the distance-based algorithm. The experimental results also show that all the grid-based algorithms showed almost equal accuracy

KEYWORDS

Location-based alert services, LBS, Location information acquisition, Distance-based acquisition algorithm.

1. INTRODUCTION

As the mobile communication technologies advance, various types of location-based services (LBS) on the wireless environments are appeared in the market. The location information of mobile device users is gathered and processed to provide the appropriate services for individuals and groups. LBS deal with peripheral information, location tracking, traffic information, location-based e-commerce, machine control, recreation, and so on [1]. LBS are on the way of development according to the diversity of users' demands. One of the LBS market analyses was released recently that according to the Gartner, Inc., in 2009 the worldwide consumer LBS subscribers and revenue will be doubled, although the mobile device sales had been dropped by 4% [2]. It also forecasts the growth of the LBS subscribers from 41.0 million in 2008 to 95.7 million in 2009, while the revenue is expected to increase from USD 998.3 million in 2008 to USD 2.2 billion in 2009.

The LBS technologies can be classified into LBS position determination, LBS platform, and LBS applications. The position determination technology is for the measurement of mobile device users' locations. The platform technology is for the servers that acquire, store and process the location data. The application technology implements various applications related to LBS for the users.

In this paper we focus on the acquisitions of location data for the location-based alert services. To provide the location-based alert services properly, an alert service system consistently observes the locations of mobile device users and alerts them when they approach and enter into or leave from the specified regions, and provides certain services previously requested by the users. Location alert services are very personalized push type services in the mobile environments. The typical location-based alert services are security services, location-based advertisement services, L-Commerce, location-based meeting/matching services, contaminated region alarm services, disaster detecting services, and logistic control services. In providing the location-based alert services, the system communication overload increases inevitably as the number of the users increases fast and so does the expense for continuously monitoring the users. Accordingly, reducing the number of user location acquisitions is a very important issue while maintaining the quality of the alert services. Several location acquisition algorithms have been propose; static acquisition algorithms [3][4], the minimum alert triggering time acquisition algorithm[5], and the distance-based acquisition algorithm [6].

In this paper we propose effective location acquisition algorithms for the location-based alert services; *the speed-based acquisition algorithm*, *the angle-based acquisition algorithm*, and *a hybrid algorithm combining the two algorithms.* The proposed algorithms are to decrease the communication overload by controlling the time interval of location acquisitions based on the movement of the users[7][8]. The speed-based acquisition algorithm adjusts the time interval based on the speed of a user; if the user moves faster then we reduce the time interval. The angle-based acquisition algorithm considers only the alert areas in the moving direction of the user for adjusting the time interval.

We further present the grid-based acquisition algorithms. They control the acquisition time interval in such a way that if a user is far away from the alert zones we set a larger value for the interval. If not, we apply each of the proposed algorithms individually. Hence we call these algorithms *the grid-speed algorithm, the grid-angle algorithm*, and *the grid-hybrid algorithm,* respectively. The experimental results showed that the speed-based, the angle-based, the hybrid algorithms reduced the average numbers of location acquisitions by 19.2%, 35.8%, and 35.6% over the distance-based algorithm, respectively. The grid-based algorithms had similar performance to their counterparts, but the grid-angle algorithm reduced the average number of acquisitions by 5.2% over the angle-based algorithm; it is about 41.0% reduction over the distance-based algorithm. All the algorithms had almost the same alert accuracy.

The rest of the paper is organized as follows. In Section 2 the location-based alert services and previous location acquisition algorithms are reviewed. In Section 3, the proposed location acquisition algorithms are introduced in detail. In Section 4 the experiment results are given. Finally Section 5 concludes the paper.

2. LOCATION-BASED ALERT SERVICES

Location acquisition means finding a user location by using mobile commutation and location determination technology. Location acquisition algorithms aim at minimizing the overhead on the network load and communication cost when acquiring location information of the users.

By efficiently controlling the time interval of location acquisitions, unnecessary location information acquisitions can be avoided. This leads to reduce the number of location acquisitions itself. Furthermore, adequately controlling the time interval also allows reducing the number of location information searches. In this section, recent location acquisition algorithms are overviewed.

2.1. The Static Location Acquisition Algorithm

The static location acquisition algorithm acquires the location information by using a fixed time interval. For all the users, the same time interval is applied for gathering their location information. In this algorithm, when the interval becomes shorter, the reliability of the services does increase, but so does the overhead of the location server. On the other hand, when the interval becomes longer, the reliability gets worse.The static algorithm is simple and easy to apply, but as the overhead of the server increases together with the increase in the number of users, the algorithm becomes not suitable for the services that might handle a large number of users.

2.2. The Minimum Alert Triggering Time Location Acquisition Algorithm

A location-based alert system *WaveAlert* controls the location search time by using two entities; MATT(minimum alert triggering time) and EAUT(earliest available update time). The maximum moving speed of the users and the distance to the nearest region (alert area) from the current location—Euclid's distance or shortest path—are used for finding a new MATT. A mobile user is guaranteed not to enter the nearest alert region at least during the MATT.

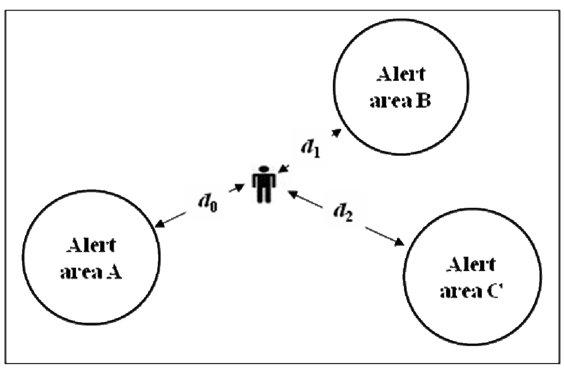


Figure 1. A user and three alert areas

Figure 1 shows that the distances between user *U* and alert areas *A, B,* and *C* are d_0, d_1, and d_2, respectively. If the maximum moving speed of *U* is $V_{\max}$, a new MATT for user *U* is $d_{shortest}/V_{\max}$, where $d_{shortest}$ is the distance between the current location of the user and the nearest alert zone, i.e, the smallest among d_i's, i=0, 1, and 2 in the figure.

However, since *WaveAlert* always uses the maximum speed of the user for obtaining the time interval, it cannot avoid 'unnecessary' location acquisitions even when the user moves at a much slower speed than the maximum speed during a considerable period of time — when the user is trapped in traffic congestion and thus does rarely move or when the user moves on foot after getting off from public transportation. EAUT denotes the users' latest location update time within the MATT period; the user's location at EAUT is used for computing d_i's (in Figure 1) so that a new MATT can be obtained. For further details, you may refer to [5].

2.3. The Distance-based Acquisition Algorithm

The distance-based acquisition algorithm dynamically controls the time interval of the location acquisition in proportion to the distance a mobile user moved and thus can be applied to the circumstances that a mobile user might move with different speeds from time to time. Controlling the time interval is performed according to the ratio of d_0 to d_1, where d_0 is the moving distance between the current location acquisition time t_0 and the previous location acquisition time t_1 and d_1 is the distance between t_1 and the location acquisition time t_2 prior to t_1. In Figure 2, locations $Loc(t_0)$, $Loc(t_1)$, and $Loc(t_2)$ represent the user's locations at times t_0, t_1, and t_2, respectively and d_0 is the shortest distance between $Loc(t_0)$ and $Loc(t_1)$ and d_1 denotes the shortest distance between $Loc(t_1)$ and $Loc(t_2)$. If $d_0>d_1$, the distance moved recently is longer, thus the time interval of the location acquisition should be reduced, and vice versa. In addition the minimum and the maximum location acquisition time intervals are also used so that the time interval should not be extremely large or small.

However, in this algorithm it is difficult to set the parameters for controlling the time interval and to set a buffer area not to trespass the alert area as shown in Figure 2. Note that the area called *the location alert buffer* that encloses a given alert area is defined for the algorithm. Right before a mobile user enters into a buffer area, the minimum time interval is used. The buffer areas work as sort of warnings to the system that the alert zones are near the users. However, if a buffer area is larger to secure the accuracy of the alert services, then unnecessary number of location acquisitions will be increased. If it is smaller, then the accuracy of the location alert services would be deteriorated.

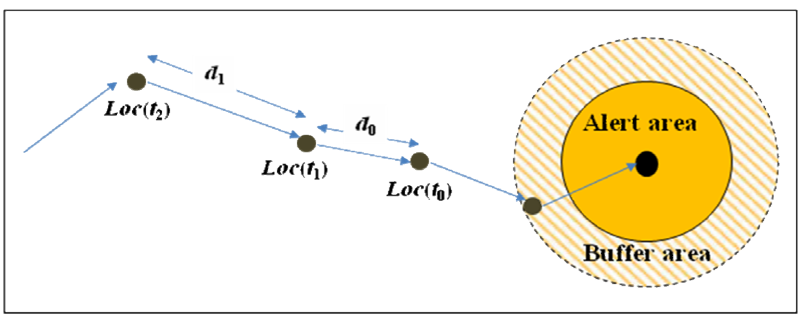


Figure 1. The distance-based acquisition algorithm

3. PROPOSED LOCATION ACQUISITION ALGORITHMS

In this section, we propose effective location acquisition algorithms.

- The speed-based acquisition algorithm
- The angle-based acquisition algorithm
- The hybrid acquisition algorithm
- The grid-based acquisition algorithms

The speed-based algorithm exploits the users' speed information. The speed-based acquisition algorithm and angle-based acquisition algorithm utilize the users' movement information to predict the future user locations and use the buffer areas as in the distance-based acquisition algorithm. The hybrid acquisition algorithm is the algorithm in which the alert areas are filtered out with the angle-base algorithm and then computes the acquisition time intervals as in the speed-based algorithm.

We also present three grid-based acquisition algorithms. Each grid-based algorithm has two phases. All the grid-based algorithms have the same first phase, while we apply the three proposed algorithms for the second phase individually. For these algorithms, we divide the input network into a grid of cells of an equal size. In the first phase, if all users are in some distance away from their own alert areas, for example, at least n cells away from their own alert areas, we set the acquisition time interval to a fixed value, which is a larger value, to reduce the number of acquisitions. Otherwise we apply each of the three proposed acquisition algorithms as the second phase.

3.1. The Speed-based Acquisition Algorithm

The speed-based acquisition algorithm uses the changes in the speed of a user. The distance-based acquisition algorithm considers only the moving distance. The distance information does not always provide proper information for computing an acquisition interval. The speed information is more appropriate for adjusting the time interval of the location acquisition, because the speed is calculated from distance as well as time. The speed-based acquisition algorithm controls the time interval in such a way that when a user is moving faster than before, the time interval is shortened and when the speed gets slower the interval is increased appropriately.

Input : the current time interval t_i, the current speed $s_{current}$, the previous speed $s_{previous,}$ and a positive constant $k < 1$, a scaling factor, is determined by the experiments
Output: a new location acquisition interval t_{i+1}

Calculate t_{i+1} as follows:
  if ($s_{previous}$ / $s_{current}$) > 1
    $t_{i+1} = t_i - k * (s_{previous} / s_{current})$
  else
    $t_{i+1} = t_i + k * (s_{previous} / s_{current})$

Algorithm 1. The speed-based acquisition algorithm

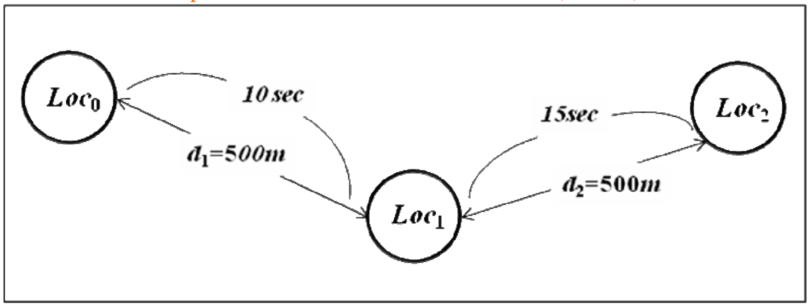


Figure 2. An example for finding a new time interval

Figure 3 gives an example that the current speed is slower than the previous one. For example, if k = 0.9 then we obtain $t_3 = t_2$ + 0.9 ((500/10)/(500/15)) = 10 + 1.35 =11.35 sec, where t_2 is 10 sec. Note that a constant k is determined by the experiments for controlling the magnitude of the value of s_{previous} / s_{current} and is set to 0.9 for all the experiments.

3.2. The Angle-Based Acquisition Algorithm

All the algorithms discussed above including the speed-based acquisition algorithm look into all the alert areas of each user for controlling the location acquisition time interval. But considering all the alert areas is a waste of the system resource, because most of the alert areas may not be entered by the user. In the angle-based acquisition algorithm, the areas that may not be entered are filtered out with the users' movement and possible moving angles. We control the time interval of the location acquisition only with these filtered alert areas.

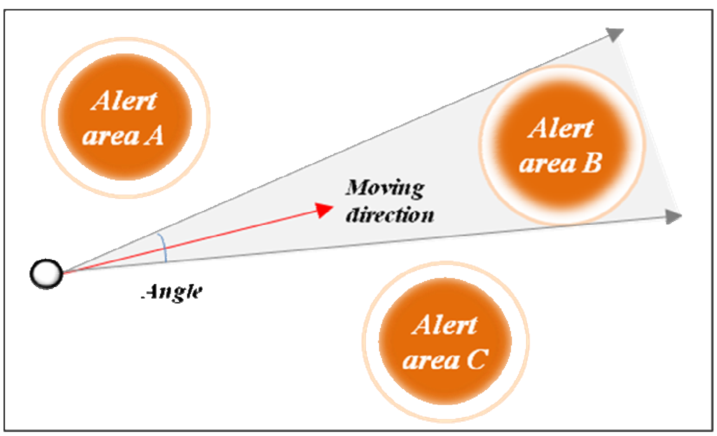


Figure 3. The concept of the angle-based acquisition algorithm

Figure 4 depicts the concept of the angle-based acquisition algorithm. We can get the user's moving direction with the user's movement information. We set the range of the moving angle to 10° after various experiments had been performed. In the figure, alert areas A and C are filtered out. The time interval for the algorithm is obtained with a well known basic physics formula written below.

distance = time * velocity + 1/2 * acceleration * time^2

The angle-based acquisition algorithm using filtered areas are described in more detail below.

Input : user speed *v,* user acceleration *a*
Output: the next location acquisition interval *t*

1. Find the alert areas in the moving direction of the user within 10° range.
2. Find the nearest alert area *Z* from the alert areas obtained in the previous step.
3. If $a = 0$, then $t = d/v$ otherwise, find *t* with solving $1/2at^2 + vt - d = 0$, where *d* is the distance between the user's location and *Z*.

Algorithm 2**.** The angle-based acquisition algorithm

3.3. The Hybrid Acquisition Algorithm

We combine the speed-based acquisition algorithm with the angle-based algorithm to get a hybrid acquisition algorithm. First, by using the concept of the angle-based algorithm, we select the alert areas for which the user heads. And then a new acquisition time interval is obtained using the speed and moving distance as in the speed-based algorithm. It is expected that the number of acquisitions is reduced with respect to the speed-based algorithm while the accuracy of the services is increased when compared with the angle-based algorithm.

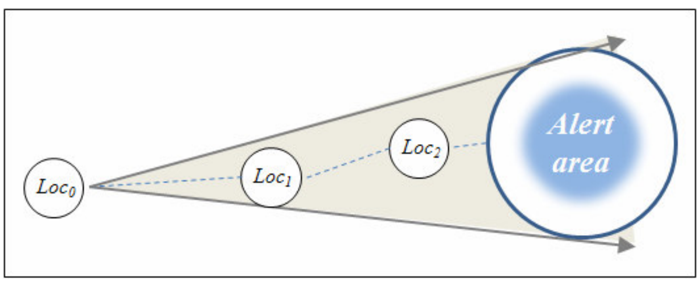


Figure 5. The concept of the hybrid acquisition algorithm

3.4. The Grid-based Acquisition Algorithms

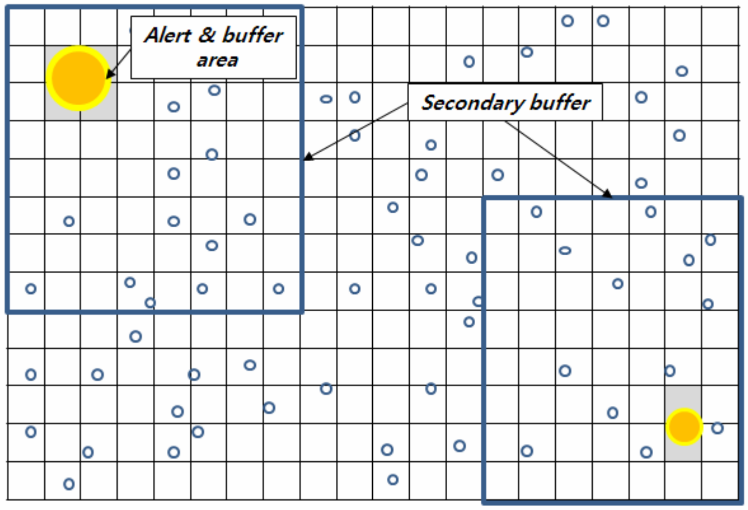


Figure 6. A network divided into a grid of cells

In this section we apply the grid-based approach to each of the proposed algorithms aforementioned. For the algorithms, we divide the input network into a grid of cells of an equal size as shown in Figure 6. The grid-based acquisition algorithms have two phases. The first phase of the algorithms is the same. In the second phase of the algorithms, each of the three proposed algorithms is applied. Hence we call these algorithms *the grid-speed algorithm, the grid-angle algorithm, and the grid-hybrid algorithm*, respectively.

In the first phase of each algorithm, we define a 'secondary' buffer area for an alert area based on the cell unit. To obtain the acquisition time interval, we first find the users who are expected to approach their own alert areas sooner than others. If these users are still some distances away from their areas — for example at least five cells away from the areas — the acquisition time interval is set to a fixed value, the maximum speed of the users / 50km, where a cell size is 10kmx10km. Note that we must determine the size of a secondary buffer appropriately, since the fixed acquisition time used in the first phase may overestimate the users' movements. Otherwise, we apply each of the proposed acquisition algorithms individually; this is the second phase. When a user entered into the 'primary' buffer area, the minimum acquisition time interval is to be applied like other algorithms.

4. EXPERIMENTS

4.1. Experimental Environment

For the experiment, *Visual Studio 2008* C++ is used. The simulation handles a total of one thousand users and the time stamp is defined from 1 to 10,000. The location data of the users are generated every five seconds and the total experiment time lasted approximately fourteen hours. In addition, the moving paths of users follow ten different scenarios, and the experiment area is 100 km * 100 km. The number of alert regions per user is set between fifteen and twenty and the size of an alert area is in the range between 1 km and 5 km. For the speed-based algorithm k is set to 0.9 throughout the experiments. Note that k is a mere scaling factor; that is to scale the value of $s_{previous}$ / $s_{current}$ down for adding to or subtracting from time t_i.

For the grid-based algorithms, each cell is a square of 10km x 10km. We had tested various sizes for the size of a secondary buffer and determined that the boundary of a secondary buffer was drawn at 5 cells away from an actual alert area. In the first phase of a grid-based algorithm, a fixed acquisition time interval was used; its value was decided as 50km/80(km/h)=225sec, where the maximum speed of the users is 80km/h and 50km came from the fact that the boundary of a secondary buffer is at 5 cells away from an alert zone. Table 1 summarizes the parameters for the experiments.

Table 1. Experimental environment

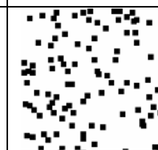
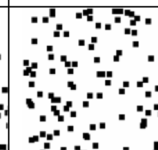
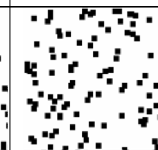
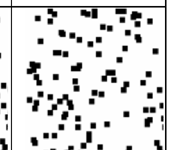
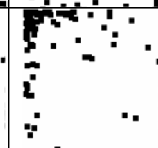
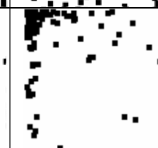
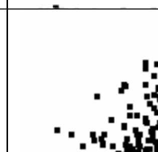
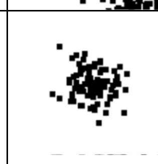
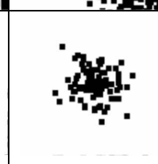
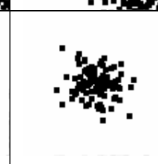
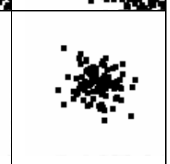
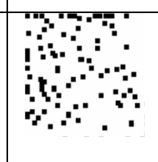
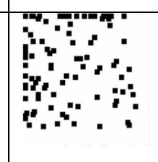
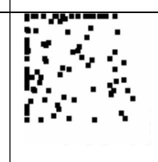
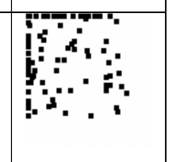
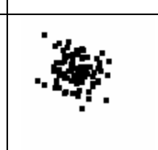
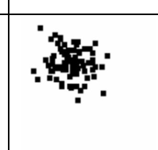
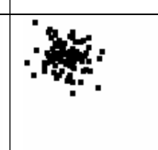
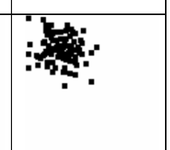
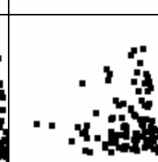
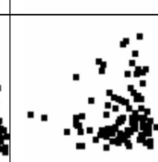
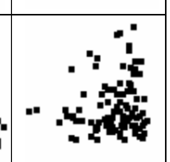
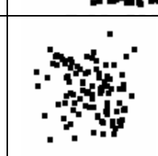
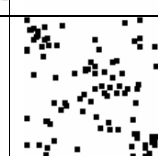
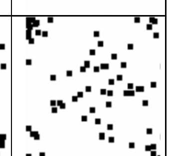
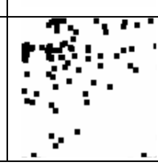
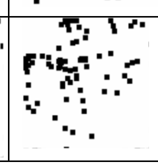
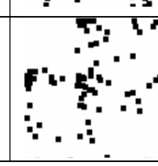
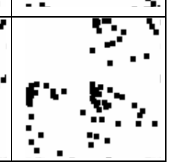
Parameter	**Value**	**Note**
number of users	1,000	
data generation interval	5 sec	
total experiment time	~14 hours	10,000x5sec ≈ 13.88hours
moving paths	10 scenario files	
area of an experiment space	100 km x 100km	
number of alert areas per user	15 ~20	
diameter of alert area	1 ~ 5km	
value of *k*	0.9	for the speed-based algorithm
size of a cell	10 km x 10 km	for the grid-based algorithms
size of the secondary buffer	5 cells	for the grid-based algorithms
acquisition time interval	225sec	for the first phase of the grid-based algorithms

4.2. Scenarios

There are ten scenarios used for the experiment according to the initial distribution methods and movement paths. They are shown in Table 2. An initial distribution allocates the starting locations of the users. We use three initial distributions; *uniform*, *skewed* and *gaussian* distributions. The moving paths of users are made according to their moving patterns as time passes by. We adopt four patterns; *uniform*, *skewed*, *3-axes*, and *all directions*. Note that for the experiment we generated ten different input files for each scenario using GSTD(generation of spatio temporal datasets)[9][10].

Table 2. Ten scenarios

Scenario	**Initial distribution**	**Moving pattern**
File 1	uniform	Uniform
File 2	skewed (northwest)	Uniform
File 3	skewed(southeast)	Uniform
File 4	gaussian	Uniform
File 5	uniform	skewed(northwest)
File 6	gaussian	skewed(northwest)
File 7	skewed(southeast)	skewed(northwest)
File 8	gaussian	all directions
File 9	skewed(northwest)	3-axes(S,SW,W)
File 10	skewed(southeast)	3-axes(N,NE,E)

Time	t_0	t_1	t_2	t_3
File 1				
File 2				
File 3				
File 4				
File 5				
File 6				
File 7				
File 8				
File 9				

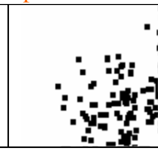
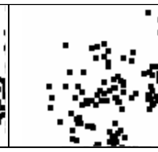
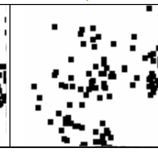
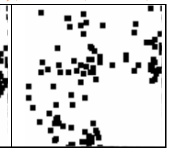
File 10				

Figure 7. The graphical views of the user distributions in the ten scenarios

4.3. Experiment Results

The average numbers of location acquisitions and alerts for the proposed algorithms and the distance-ratio acquisition algorithm have been evaluated and compared. We also developed the grid-based algorithm for the distance-ratio acquisition algorithm for comparison and call it *the grid-distance algorithm*. Other algorithms — the static and the MATT algorithms — are not compared, since the distance-based algorithm outperformed these algorithms.

The experiments have been performed to measure two factors. First, we find the *success ratio* that is the percentage of the actual alerts issued within the alert areas to the total number of alerts. Second, the *number of acquisitions* of each algorithm was measured. If an algorithm has a higher success ratio with a smaller number of acquisitions, it would be the best choice in practice.

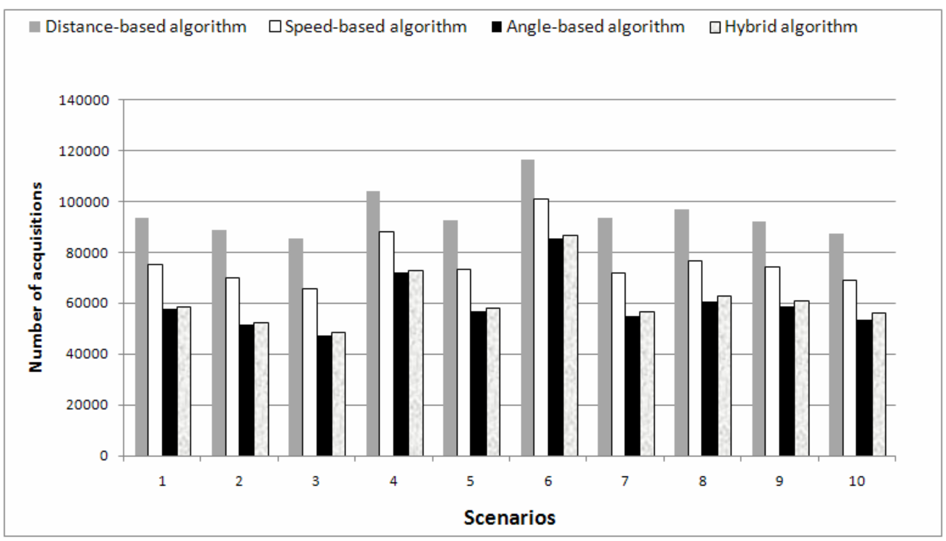


Figure 8. Average numbers of location acquisitions of the algorithms without applying the grid-based approach

Figure 8 compares the average numbers of location acquisitions for the distance-based acquisition algorithm and the proposed algorithms except the grid-based algorithms. For each scenario file, the angle-based algorithm showed the best performance and the speed-based algorithm outperformed the distance-based algorithm. The speed-based algorithm, the angle-based algorithm, and the hybrid algorithm showed average of 19.2%, 35.8%, and 35.6% reduction in the number of location acquisitions over the distance-based algorithm, respectively. Such reductions were possible since the proposed algorithms take advantage of the speeds of users and the angle-based algorithm utilizes the moving directions of users. The hybrid algorithm could not perform better than the angle-based algorithm, because the acquisition time intervals of the hybrid algorithm were computed with the same method in the speed-based algorithm. In the experiments, we found that the intervals of the hybrid algorithm were shorter than those of the angle-based algorithm.

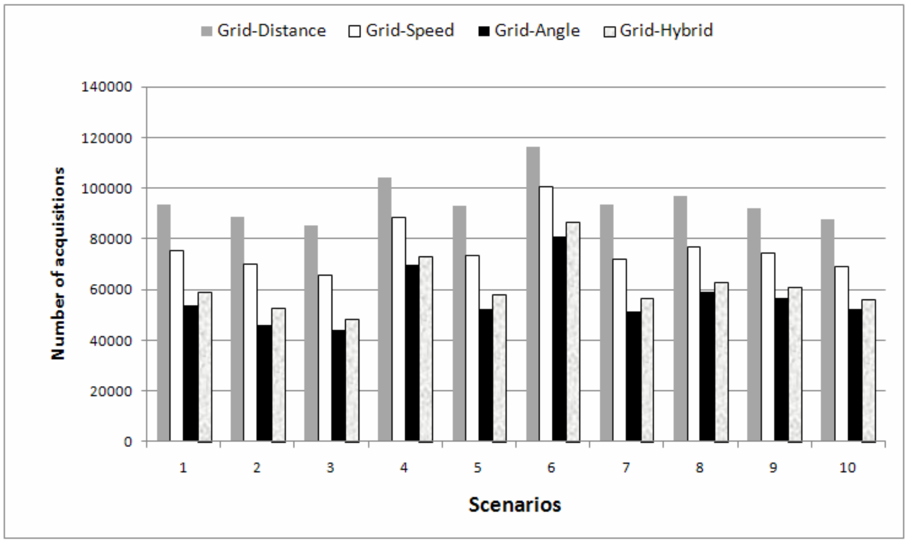


Figure 9. Average numbers of location acquisitions for the grid-based acquisition algorithms

Table 3. Numbers of location acquisitions for the ten scenarios

Algorithm / *Scenario*	Distance-based	Speed-based	Angle-based	Hybrid	Grid-Distance	Grid-Speed	Grid-Angle	Grid-Hybrid
File 1	93519	75464	57929	58846	93457	75450	53621	58878
File 2	88903	69784	51762	52702	88888	69784	46035	52715
File 3	85439	65860	47433	48463	85295	65848	44101	48542
File 4	104026	88299	72290	73153	104026	88299	69603	73153
File 5	92907	73219	56667	58210	92868	73217	52339	58217
File 6	116446	100791	85290	86704	116446	100791	81013	86704
File 7	93635	72139	54866	56579	93529	72129	51265	56589
File 8	96916	76586	60898	63099	96859	76603	59097	63140
File 9	92103	74140	58692	61132	92091	74163	56786	61153
File 10	87573	68820	53726	56279	87511	68824	52512	56291
Average improvement over the distance-based algorithm	0.0%	19.2%	35.8%	35.6%	0.1%	19.2%	41.0%	35.6%

Figure 9 compares the average numbers of location acquisitions for the grid-based acquisition algorithms. The graph shows that the grid-angle algorithm(*Grid-Angle*) has the best performance. The angle-based algorithm, the hybrid algorithm, and the grid-hybrid algorithm(*Grid-Hybrid*) showed similar performances. The grid-distance algorithm(*Grid-Distance*) could hardly improve the number of acquisitions. But Grid-Angle improved 5.2% over the angle-based algorithm while other grid-based algorithms did not improve their counterparts. Table 3 shows the numerical data for Figure 9.

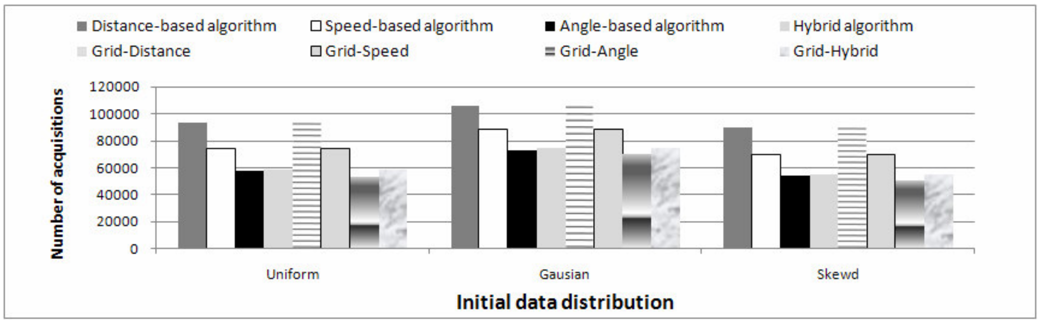


Figure 10. Average numbers of location acquisitions for different data distributions

Figure 10 illustrates the results of the experiment based on different initial data distributions. Regardless of all the data distribution types, Grid-Angle showed the best results showing the robustness.

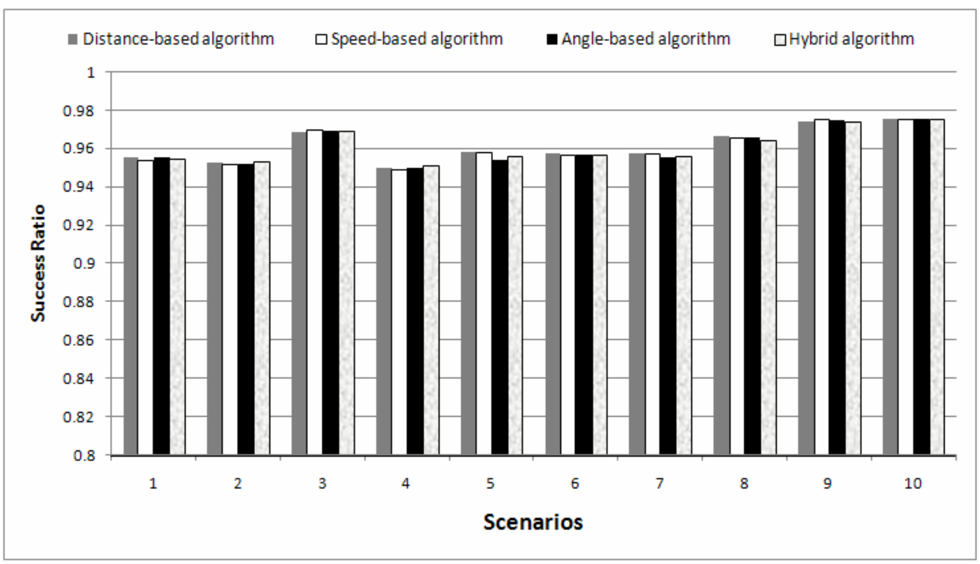


Figure 11. Success ratios of the algorithms without applying the grid-based approach

Figure 11 compares the success ratios for the algorithms without applying the grid-based approach. As shown in the figure, all four algorithms showed similar levels of the success ratios for all the scenarios, because all the algorithms used primary buffer areas. The hybrid algorithm showed the best success ratio on the average. The average number of alerts for each scenario is different from each other, because each has a different pair of initial distribution and moving pattern. These results proved that the proposed algorithms do not deteriorate the level of accuracy performance while reducing the number of location data acquisitions effectively. Figure 12 shows the success ratios for the grid-based acquisition algorithms.

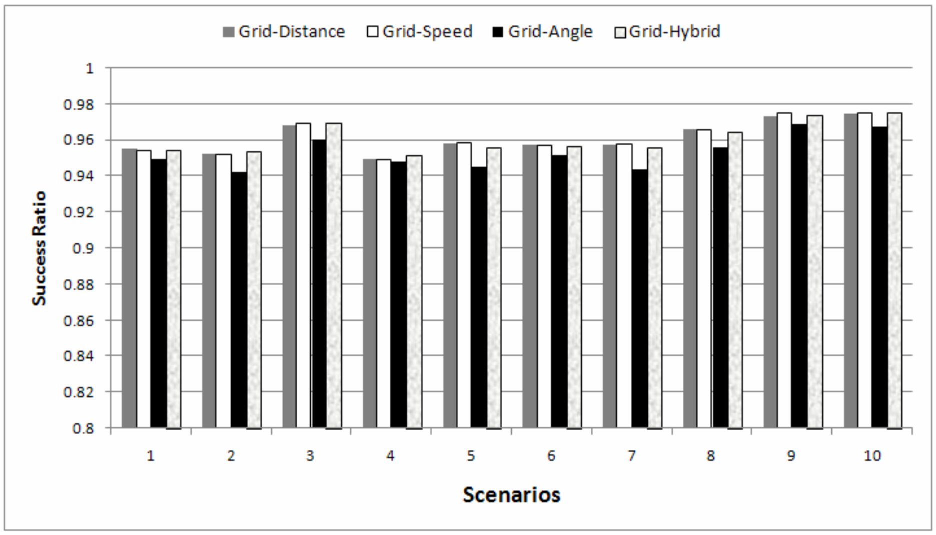


Figure 12. Success ratios of the grid-based acquisition algorithms

Table 4. The average success ratios for the algorithms

Algorithm / ratio	*Distance-based*	*Speed-based*	*Angle-based*	*Hybrid*	*Grid-Distance*	*Grid-Speed*	*Grid-Angle*	*Grid-Hybrid*
average success ratio	0.957	0.957	0.956	0.957	0.957	0.957	0.949	0.957

Table 4 summarizes Figures 11 and 12 numerically. As shown in the table, all the algorithms except Grid-Angle had almost the same average success ratios. The average success ratio of Grid-Angle was 0.8% smaller than others, because sometimes longer time intervals than the predefined time interval were used. Such cases may occur when a user is not near the secondary buffer area and moves fast. The algorithm may overestimate the interval so that when the location server acquires the location data of the user at the next location acquisition time, the user has already entered into the alert area before an alert message is issued.

5. CONCLUSION

A major drawback of the distance-based acquisition algorithm is revealed from the fact that it simply considers the user's moving distance. Although the user's moving distance is increased during a long period of time, it does not necessarily mean that the user moved with a faster speed. In this case, however, the distance-based algorithm regards the user's moving speed to be faster and hence reduces the time interval of the location acquisition. This induces an increase in the number of location acquisitions. The speed-based acquisition algorithm reduces the number of location acquisitions in this case, because it utilizes the speed.

In this paper the angle-based acquisition algorithm has also been proposed. It considers users' moving direction and hence reduces the number of unnecessary location acquisitions further. If there is no alert region in the direction of users' moving direction, the algorithm does not reduce the time interval even the moving speed is accelerated. Both proposed algorithms showed improved performances while they both maintain the same level of accuracy. The hybrid acquisition algorithm showed a better average number of acquisitions over the speed-based algorithm, but did not beat the angle-based algorithm. The acquisition time intervals of the hybrid algorithm were determined with the same method in the speed-based algorithm; the intervals were little bit longer than those of the angle-based algorithm. Hence the average number of acquisitions of the hybrid algorithm is little more than that of the angle-based algorithm.

We applied the grid-based approach to each of the proposed algorithms for further reducing the number of location acquisitions. In the first phase of a grid algorithm, the location server collects a fewer amount of location data by using a longer acquisition time interval. The average number of location acquisitions of the grid-angle acquisition algorithm was improved by 5.2% over the angle-based algorithm. However, other grid-based algorithms hardly improved over their counterparts, because other algorithms, regardless of the location of the current cell where a user is located, determine the acquisition time intervals by considering all the alert areas of the user. The accuracies of the algorithms we tested were almost the same except the grid-angle algorithm. But the difference between the accuracy of others and that of the grid-angle algorithm was only 0.8%. In conclusion, the grid-angle algorithm is the best choice among the algorithms for the practical environments. As our future work, we plan to study on developing efficient location search and control algorithms for the cases where group users are involved while the areas are changed dynamically.

ACKNOWLEDGEMENTS

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) for the research (2009-0073072).

REFERENCES

[1] U. Varshney, (2001) "Location Management Support for Mobile Commerce Applications", *International Conference on Mobile Computing and Networking*, pp. 1-6.

[2] http://www.mobilemonday.net/news/lbs-market-will-double-in-2009

[3] K. Min & J. Park, (2003) "Techniques for Acquisition of Moving Object Location in LBS", *Proceedings of FIG Conference*, pp 1-14.

[4] A. Bar-Noy, I. Kessler & M. Sidi, (1995) "Mobile Users: to Update or Not to Update?", *Wireless Networks*, Vol. 1, No. 2, pp. 187-196.

[5] Wave Market Incorporated, (2004) "Mobile Tele Communication Network System That Provides Moving Objects with Alert-Based Service", *Official Patent Publication in Korea Republic*, Application No. 10-2004-7000651.

[6] B. Ahn, S. Yang, H. Jin & J. Lee, (2005) "Location Poling Algorithm for Alerting Service Based on Location", *Springer Verlag Lecture Notes in Computer Science,* Vol. 3833, pp. 104-114.

[7] Dieter Pfoser & Christian S. Jensen, (1999) "Capturing the Uncertainty of Moving-Object

Representations", *Proceedings of 6th International Symposium on Advances in Spatial Databases*(*SSD*), pp.111-132.

[8] Ouri Wolfson, S. Chamberlain, S. Dao & L. Jiang, G. Mendez, (1998) "Cost and Imprecision in Modeling the Position of Moving Objects", *Proceedings of the Fourteenth International Conference on Data Engineering* (ICDE14).

[9] R. Güting, D. Papadias & F. Lochovsky, (1999) "On the Generation of Spatiotemporal Datasets", *Springer Verlag Lecture Notes in Computer Science*, Vol. 1651, pp. 147-164.

[10] Y. Theodoridis & M. Nascimento, (2000) "Generating spatiotemporal datasets on the WWW", *ACM SIGMOD Record*. Vol. 29, No. 3, pp. 39-43.