Spatial Alarms In Obstructed Space*

Tanzima Hashem[†]
Department of Computer Science
Bangladesh University of Engineering and Technology
Dhaka,Bangladesh
tanzimahashem@gmail.com

Md Touhid Zaman[‡]
Department of Computer Science
Bangladesh University of Engineering and Technology
Dhaka,Bangladesh
tz08128@gmail.com

Sezana Fahmida[§]
Department of Computer Science
Bangladesh University of Engineering and Technology
Dhaka,Bangladesh
sezanafahmida@gmail.com

ABSTRACT

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

The high availability of smart phones has led to the proliferation of location based services. According to many, the next step in location based services is Spatial Alarms. Many believe this particular feature is going to dominate the future mobile-phone computing systems. Spatial alarms are an extension of time-based alarms. It is, however, triggered by a specific location irrespective of time."Remind me if I'm within 100 meters of a pharmacy" is a possible example of a spatial alarm. It is a personalized location based service which can vary from user to user. Existing research has categorized spatial alarms into three types: public, shared and private. Public alarms are alarms which are active for every user within the system, such as an alarm must be sent to everyone within 100 meters of a building on fire. Private alarms are user defined alarms which are only viewable by the user herself, such as a user might set an alarm to alert her if she is within 100 meters of her favourite coffee shop. Shared alarms are shared between specific groups of people. In the previous example if a user chooses to share the alarm for the coffee shop with some of her friends it becomes a shared spatial alarm.

It is noteworthy that spatial alarms are quite dissimilar to spatial range query. Spatial alarms are based on a fixed location thus applying the techniques that are used in answering spatial range queries is both inefficient and wasteful for the two dominating reasons, Firstly, in spatial range query as the user is the main point of interest, continuous re-evaluation of her location is needed in case of a mobile user. In contrast, spatial alarms need only be re-evaluated when the user in approaching a specific location. Secondly, in spatial alarm, the main point of interest is a static location. So the user's location is not relevant at all times. It is quite clear that applying spatial range query techniques in evaluating spatial alarms is going to result in wastage of resources. If we start to evaluate spatial alarms as soon as the user is on the move even if the user is far away from her desired location our efforts will be futile.

Existing work in this area has focused mainly on Euclidean distance and road network models, to the best of our knowledge; no work is yet done on spatial alarms in obstructed space. Spatial alarm evaluation in obstructed space is different than road network or Euclidean space as it considers the obstacles in the path to the location of alarm. It is better approximated by a pedestrian scenario while road networks are approximated by vehicle scenarios. A vehicle can only go to a specific location following a predefined road, but a pedestrian's path is not limited by roads. However, a pedestrian is obstructed by various obstacles such as buildings or trees. So while calculating the distance from spatial alarms, we have to consider the obstructed distance.

Spatial alarms are location-based, user-defined triggers which will possibly shape the future mobile application computations. They are distinct from spatial range query and do not need immediate evaluation after the user has activated them.The spatial alarm evaluation strategies are judged based on two features, High accuracy and High system scalability. High accuracy refers to the quality that guaranties no alarms are missed. And High scalability is the feature that ensures that the system can adapt to a large number of spatial alarms. In this paper, We propose a novel approach to evaluate spatial alarms in obstructed space which ensures

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

†Dr. Trovato insisted his name be first.

‡The secretary disavows any knowledge of this author's actions.

§This author is the one who did all the really hard work.

both high accuracy and high scalability.

2. PROBLEM SETUP

Obstructed Space Route Problem denotes the problem of finding the shortest route between two query-points in Obstructed Space where non-intersecting 2D polygons represent *obstacles* and where the route does not traverse through any obstacles.The length of the Obstructed route between points a and b is called the *Obstructed Distance* between a and b, denoted by $D_{obs,a,b}$.
A **Spatial Alarm Query in Obstructed Space** is formally defined as follows:

Definition 2.1. Given the user's current location p,and an alarming distance U_d for an alarm, spatial alarm query returns the set of alarms A, where for each a$\in$ A,$D_{obs,p,a}$<U_d.

We define three different type of regions: *Known Region*,*Reliable Region* and *Safe Region*

Definition 2.2. Known Region: We define two different known region for the POIs and the obstacles.The region containing at least 1 POI is the known region for POI.
The region circulating the POIs known region containing none or single colliding obstacles is the known region for the obstacles.The set of obstacles and POIs within this region is known to the client. We will denote the radius of the POIs known region as r_{kp} and that of the obstacles known region as r_{ko}

Definition 2.3. Reliable Region: We will denote the radius of the reliable region as $R_{reliable}$.Given the user's previous location P_1 and current location P_2 if, P_1-P_2<$R_{reliable}$, then no further queries to the server has to be done to compute a consistent set of answers.

Definition 2.4. Safe Region: The region located inside reliable region within which the answer set of POIs remains unchanged for a moving client.We will denote the radius of the safe region as R_{safe}.Given the user's previous location P_1 and current location P_2 if, P_1-P_2<$R_{reliable}$, then no recalculation is needed to compute the answer.

3. RELATED WORK

Extensive research has been performed and various effective algorithms have been proposed [4],[5],[1] to process spatial alarms in Euclidean space and road network in recent years. Euclidean space considers the straight line distance between two points irrespective of obstacles on the other hand in road networks navigation is limited along predefined roads.
Obstructed space considers the shortest distance between two points in the presence of obstacles. Various spatial range query algorithms have been presented in recent times [7],[3],[6] such as nearest neighbour and group nearest neighbour in obstructed space.
Again, comprehensive research [2] has been conducted to make spatial alarm evaluation energy-efficient and effective in road networks. However, to the best of our knowledge no research work has yet been published on the topic of spatial alarms in obstructed space.

4. SPATIAL ALARMS IN OBSTRUCTED SPACE

Our spatial alarm evaluation system is divided into client-server architecture. The server has access to the locations of mobile users, location of alarms and location of obstacles. In this paper we assume that all users have access to some sort of localization service such as GPS or Wi-Fi that allow the server to pinpoint their current location. The client application is a thin-weight application that communicates with the server at regular intervals to retrieve necessary information about alarms and the obstacles. With the help of the client application the users can personalize their alarms as public, private or shared. They can join any public alarm or shared alarm as per their choice. We assume that the user can use any device such as smart-phones or PDA.
Our aim is to provide an energy-efficient, concise and accurate spatial alarm service for obstructed space. To preserve energy, the popular approach is to put the device in use to a low-power consumption state or sleep state. We propose a novel approach of computing a *safe region.* As long as the client device is in the safe region, the device can be put into sleep state without the risk of any alarm being missed.

Spatial alarm evaluation can be optimized using two key features: firstly, reducing the number of device wake-ups and second reducing the re-computation of same obstructed distance and reducing the number of duplicate data retrieval from the server. For the first strategy to be successful our safe-region computation should be accurate and optimal. We propose an algorithm in the section which will compute an optimal safe-region for our spatial alarm evaluation system. The second optimization technique is related to the safe-region computation technique. To compute the safe region the client application must communicate with the server as it needs the location of obstacles and alarms. In this paper we aim to optimize this communication by ensuring that no redundant data is retrieved from the server. We propose two different type of strategies which highlight exactly one of the aformentioned key features. Our application has two different modes,namely, *Bandwidth Saving Mode* and *Computational Cost Saving Mode*

4.1 Bandwidth Saving Mode

In this mode the main focus is to reduce the bandwidth of communication between the server and the client.In this mode the procedure is divided into three steps. *Initialization,computing the reliable region and the safe region* and *Update on location change.* Algorithm 1 shows the pseudocode for initialization. The input to the algorithm is the location of the client, the outputs of the algorithms are radius of the known regions (r_{kp},r_{ko}),the set of all obstacles within r_{ko} and the set of all POIs within r_{kp}.

4.2 Computational Cost Saving Mode

5. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LaTeX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

Algorithm 1 Initialization

```
1: procedure INITIALIZATION
2: incrementally increase r_k p untill it finds atleast 1 POI
3: Make P = set of all POI inside the circular region of radius r_k p
4: Retrieve all the obstacles inside the circular region of radius r_k p
5: r_k o = r_k p
6: Construct the visibility graph V_G for the POIs
7:     while there is any unreachable POI in the V_G and there are more than 1 collision of obstacle perimeter of the circle of radius r_k o do
8:         expand rko until the unreachable POI gets reachable or there is less than 2 collisions
9:         retrieve new obstacles;
10:        If any new POI is found, then let r_k p = r_k o and go to the 3rd step
11:    end while
12: Make O = set of all obstacles inside the radius r_k o
13: Return the client a response with r_k p, r_k o, V_G, P, O
14: end procedure
```


6. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

7. REFERENCES

[1] Bhuvan Bamba, Ling Liu, Arun Iyengar, and Philip S. Yu. Distributed processing of spatial alarms: A safe region-based approach. In *29th IEEE International Conference on Distributed Computing Systems (ICDCS 2009), 22-26 June 2009, Montreal, Québec, Canada*, pages 207–214. IEEE Computer Society, 2009.

[2] Myungcheol Doo, Ling Liu, Nitya Narasimhan, and Venu Vasudevan. Efficient indexing structure for scalable processing of spatial alarms. In Divyakant Agrawal, Pusheng Zhang, Amr El Abbadi, and Mohamed F. Mokbel, editors, *18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2010, November 3-5, 2010, San Jose, CA, USA, Proceedings*, pages 426–429. ACM, 2010.

[3] Yunjun Gao, Jiacheng Yang, Gang Chen, Baihua Zheng, and Chun Chen. On efficient obstructed reverse nearest neighbor query processing. In Isabel F. Cruz, Divyakant Agrawal, Christian S. Jensen, Eyal Ofek, and Egemen Tanin, editors, *19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2011, November 1-4, 2011, Chicago, IL, USA, Proceedings*, pages 191–200. ACM, 2011.

[4] Kisung Lee, Emre Yigitoglu, Ling Liu, Binh Han, Balaji Palanisamy, and Calton Pu. Roadalarm: A spatial alarm system on road networks. In Christian S. Jensen, Christopher M. Jermaine, and Xiaofang Zhou, editors, *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, pages 1372–1375. IEEE Computer Society, 2013.

[5] Anand Murugappan and Ling Liu. An energy efficient middleware architecture for processing spatial alarms on mobile clients. *MONET*, 15(4):543–561, 2010.

[6] Nusrat Sultana, Tanzima Hashem, and Lars Kulik. Group nearest neighbor queries in the presence of obstacles. In Yan Huang, Markus Schneider, Michael Gertz, John Krumm, and Jagan Sankaranarayanan, editors, *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Dallas/Fort Worth, TX, USA, November 4-7, 2014*, pages 481–484. ACM, 2014.

[7] Jun Zhang, Dimitris Papadias, Kyriakos Mouratidis, and Manli Zhu. Spatial queries in the presence of obstacles. In Elisa Bertino, Stavros Christodoulakis, Dimitris Plexousakis, Vassilis Christophides, Manolis Koubarakis, Klemens Böhm, and Elena Ferrari, editors, *Advances in Database Technology - EDBT 2004, 9th International Conference on Extending Database Technology, Heraklion, Crete, Greece, March 14-18, 2004, Proceedings*, volume 2992 of *Lecture Notes in Computer Science*, pages 366–384. Springer, 2004.

APPENDIX

A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 The Body of the Paper

A.2.1 Type Changes and Special Characters

A.2.2 Math Equations

Inline (In-text) Equations.

Display Equations.

A.2.3 Citations

A.2.4 Tables

A.2.5 Figures

A.2.6 Theorem-like Constructs

A Caveat for the TEX Expert

A.3 Conclusions

A.4 Acknowledgments

A.5 Additional Authors

This section is inserted by LaTeX; you do not insert it. You just add the names and information in the `\additionalauthors` command at the start of the document.

A.6 References

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the [5] .bbl file. [4] Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.

B. MORE HELP FOR THE HARDY

The sig-alternate.cls file itself is chock-full of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of LaTeX, you may find reading it useful but please remember not to change it.