Viewer of Synthetic Scenarios to Evaluate Estimators of Local Coverage Areas Based on Detected Objects

Jorge Bernad
University of Zaragoza
Zaragoza, Spain
jbernad@unizar.es

Carlos Bobed
University of Zaragoza
Zaragoza, Spain
cbobed@unizar.es

Eduardo Mena
University of Zaragoza
Zaragoza, Spain
emena@unizar.es

ABSTRACT

We present a visual tool to test different algorithms that estimate the real coverage area of a certain (mobile or static) object against previously generated scenarios. Such algorithms consider as input data just the location of the devices that can communicate with that object.

CCS CONCEPTS

• **Information systems → Mobile information processing systems**; **Location based services**; **Evaluation of retrieval results**;

KEYWORDS

Coverage Area Estimation, Testing Location Based Services

1 INTRODUCTION

Knowing the actual area that a (static or mobile) object is able to scan from its current location is still an open problem, but it is needed to know whether a given interesting area is completely monitored by a set of active (mobile) monitors. Although some approaches assume a simple model like a disk of a certain radius centered on the location of objects [4] and others use extra geographic information and define complex physical models of the behavior of wireless signals [2, 3], these models do not fit well in any real situation, and/or require information that is not always available.

In this paper we present a tool to execute and evaluate different strategies that estimate the coverage area of an object in different generated scenarios, with a varying number of objects and obstacles. This tool provides us with visual feedback of the outcome, allowing us to evaluate different estimators of the coverage area of a moving object that take as basis the location of objects that such a device detects from its current location at each time. The synthetic scenarios range from wide open to narrow paths surrounded by multiples obstacles, with a selectable number of objects detected by the reference device. With our tool the behavior of coverage area algorithms can be quickly and easily observed (visually and analytically) under many different situations before testing them in real scenarios.

2 COVERAGE AREA ESTIMATORS

In this section, we briefly present the coverage area estimation algorithms implemented in our prototype, although other could be added.

MobiQuitous 2017, November 7–10, 2017, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5368-7/17/11…$15.00
https://doi.org/10.1145/3144457.3144517

Disk Method. This broadly adopted [4] and simple estimator assumes that the coverage area is a disk of a fixed radius (based on the theoretical maximum range of the wireless communication technology used) centered on the location of objects.

Adaptive Disk Method. This estimator is a variation of the disk model where the coverage area is assumed to be a disk whose radius r is the distance to the furthest visible object in the scenario.

Convex Hull Method. This method is based on the recursive subdivision of the triangulation given by the convex hull of the positions of the detected objects. The subdivision criteria depends on a value α_i calculated for each candidate triangle, which in turns depends on a value A that represents the size of an "*empty big area*". Such A value is obtained detecting area outlier triangles.

Polar Grid Method. This method is based on a polar grid (a grid in polar coordinates of the points in the plane). It is determined by a maximum distance ρ, and the number of divisions r and s for the module and the angle. Intuitively, the method avoids the *obstacles* in the real coverage area by means of the propagation (with mitigation) of the witnesses' votes towards the center of the grid.

Delaunay Triangulation Method. This method, which is also aware of the presence of obstacles, is based on obtaining the *Delaunay triangulation* of the detected objects. After having the triangulation, this method filters out big empty areas exploiting the fact that the Delaunay triangulation maximizes the area of the triangles regard to its perimeter.

Mixed Method. Finally, based on an off-line evaluation, we include an algorithm which just selects the estimator that has shown the best behavior depending on the amount of detected objects. That is, it is a pre-trained wrapper of the previous algorithms.

3 ALGORITHM VIEWER

We have developed an application to automatically generate synthetic scenarios of different nature, which are used in our visual tool to test different coverage area estimators. The reason for generating the scenarios a priori is to make the scenario tests be repeatable.

3.1 Off-line Generation of Synthetic Scenarios

We have automatically generated a set of 800 synthetic scenarios where the coverage area is represented by a disk of 300 distance units with a variability of ±5% in different directions[1], centered on the location of the so called *reference object*. This ideal (disk) coverage area of each scenario is covered with *obstacles* up to a 90%, resulting in 10 different groups of 80 scenarios ranging from areas with no obstacles to heavily obstacle-covered ones ([80%-90%] of the area covered by *obstacles*).

[1] We adopted a dimensionless quantity for the sake of generality, but the values are based in WiFi distances and powers considered in [2].

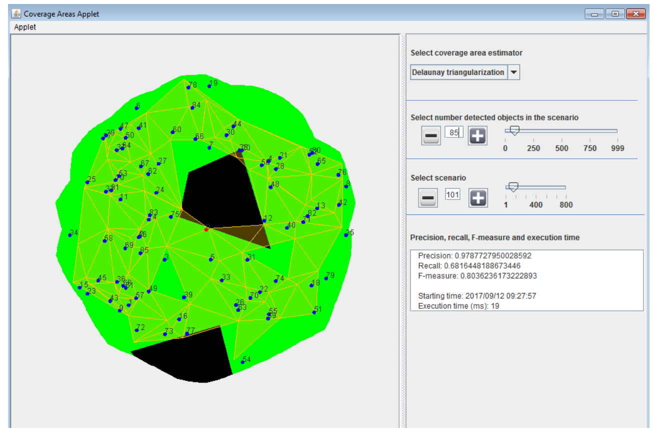

Figure 1: Delaunay-based algorithm estimation for #101 scenario and 85 detected objects.

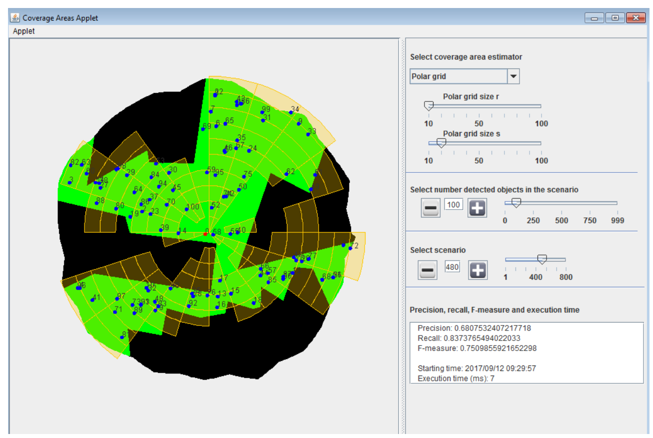

Figure 2: Polar grid-based algorithm estimation with a 10×20 **grid for #480 scenario and 100 detected objects.**

Each scenario contains a set of detected objects (ranging from 0–999) randomly placed within the coverage area. Their position represent a snapshot of the detected objects from the reference object which is in the center of the scenario.

3.2 Viewer Graphical User Interface

We have implemented a prototype[2] using Java 1.8 as programming language, and the JTS Topology Suite 1.14 to perform all the geometrical operations.

The prototype shows an scenario (where obstacles are presented in black and the real coverage area in green) as seen by the reference object (in red), which appears near the center of the scenario. In Figures 1 and 2 we show the execution of Delaunay and Polar Grid algorithms on different scenarios. The selected scenario, number of objects, and graphical representation of the estimated coverage area around the reference object are visualized on the left panel, for the parameters selected on the right panel, which are the following:

- *Selection of estimator*: We can select among different estimation algorithms. For some algorithms we can specify their specific parameters: α threshold parameter (range 0-1) for Convex Hull; and radius grid size r (range 10-100) and angular grid size s (range 10-100) for Polar Grid.
- *Number of objects*: We can specify the number of object detected by the reference object, ranging from 0-999. Indeed, the more detected objects the best results (in general) that the different algorithms obtain, although in real life it is usually a low number.
- *Scenario*: The application allows us to select among 800 synthetic scenarios previously generated as explained before, with different topologies, and number and location of obstacles.
- *Precision, recall, F-measure and execution time*: The well-known measures and the execution time are calculated and presented each time a parameter changes in the GUI. The precision of a coverage area estimator is defined as the percentage of the estimated area which belongs to the real coverage area (i.e., it is correctly estimated), and the recall is defined as the percentage of the real coverage area which is correctly estimated.

In order to include uncertainty about the actual location of the objects, the actual position of each object is considered to be within a disk centered in the obtained location and of a radius of 10% of the coverage area radius, what it is called *the uncertainty location area* [1] of an object. We obtain this uncertainty value taking into account the work in [2]: 1) we consider that the location beacons are sent every 3 seconds, 2) we have a maximum speed of the objects of 5.5 m/s (above regular people jogging, 3.33 m/s), and 3) we assume a maximum error of 5 meters in the location method that each device uses. This leads us to a maximum displacement of 21.5 meters, 7% of the maximum radius obtained in their experiments. Our prototype considers such an uncertainty in every refreshment of the GUI, so the same algorithm retrieves slightly different results every time due to the jitter introduced in the locations.

4 CONCLUSIONS AND FUTURE WORK

As moving object trajectories and map information could be unavailable, we need algorithms that are able to estimate the coverage area in the absence of such an information. In this context, we have developed a tool that automatically generates scenarios, and a scenario viewer that allows developers to evaluate visually and analytically their estimation algorithms before going to real deployments. We plan to extend our web-based viewer to allow online users to provide their own algorithms.

[2]Our prototype is available as a Java Applet and a Java Web Start application at http://sid.cps.unizar.es/SHERLOCK/coverageAreas/.

Acknowledgments. This research work has been supported by projects TIN2013-46238-C4-4-R, TIN2016-78011-C4-3-R (AEI/FEDER, UE), and DGA/FEDER.

REFERENCES

[1] J. Bernad*, C. Bobed*, S. Ilarri, and E. Mena. 2017. Handling Location Uncertainty in Probabilistic Location-Dependent Queries. *Information Sciences* 388–389 (May 2017), 154–171.
[2] S. D. Hermann, M. Emmelmann, O. Belaifa, and A. Wolisz. 2007. Investigation of IEEE 802.11k-based Access Point Coverage Area and Neighbor Discovery. In *32nd IEEE Conference on Local Computer Networks (LCN'07)*. 949–954.
[3] J. Robinson, R. Swaminathan, and E. Knightly. 2008. Assessment of Urban-scale Wireless Networks with a Small Number of Measurements. In *14th ACM Intl. Conf. on Mobile Computing and Networking (MobiCom'08)*. 187–198.
[4] R. Verdone, D. Dardari, G. Mazzini, and A. Conti. 2010. *Wireless Sensor and Actuator Networks*. Academic Press.