2014 Canadian Conference on Computer and Robot Vision

The Range Beacon Placement Problem for Robot Navigation

River Allen, Neil MacMillan, Dimitri Marinakis, Rahnuma Islam Nishat, Rayhan Rahman, and Sue Whitesides
Computational Geometry and Robotics Lab (CGAR),
Department of Computer Science,
University of Victoria, Canada
{riverallen, neilrqm, dmarinak}@gmail.com, {rnishat, rayhan, sue@}uvic.ca

***Abstract*—Instrumentation of an environment with sensors can provide an effective and scalable localization solution for robots. Where GPS is not available, beacons that provide position estimates to a robot must be placed effectively in order to maximize a robots navigation accuracy and robustness. Sonar range-based beacons are reasonable candidates for low cost position estimate sensors. In this paper we explore heuristics derived from computational geometry to estimate the effectiveness of sonar beacon deployments given a predefined mobile robot path. Results from numerical simulations and experimentation demonstrate the effectiveness and scalability of our approach.**

***Keywords*-Localization, acoustic ranging beacons, roomba, hill climbing, simulated annealing, random maps**

I. Introduction

We consider the problem of positioning a set of range-only beacons in an environment to facilitate the navigation of a mobile robot. We assume that the robot can determine an imprecise range estimate to a subset of these beacons and also that the robot can use odometry information to aid its position and orientation (or pose) estimate. Given a robot with only these sensing capabilities and a specified path within a given environment map, we ask if there is a beacon deployment that allows the robot to navigate the path with minimal risk, that is, acceptable deviation from the desired path.

Using computational geometry, we introduce a novel set of heuristics that allow us to quantify the merit of a given beacon deployment for a given path. Having a utility estimate for a particular beacon deployment, we can then search over the space of beacon assignments to find a deployment that has high utility according to our measure. For a given beacon deployment, our approach assigns each region in the environment a *localization utility* based on an estimate of the positional uncertainty of the robot should its true location be in that region. We then integrate the localization utility along a specified path to evaluate the suitability of a proposed beacon pose.

In this paper we make the following contributions: 1) a pragmatic formulation of the beacon localization problem using geometric primitives; and 2) an investigation into scalable search methodologies using this geometry-based formulation. Our combined application of computational geometry and discrete state space search methods leads to a scalable solution to the beacon placement and orientation problem in mobile robotics.

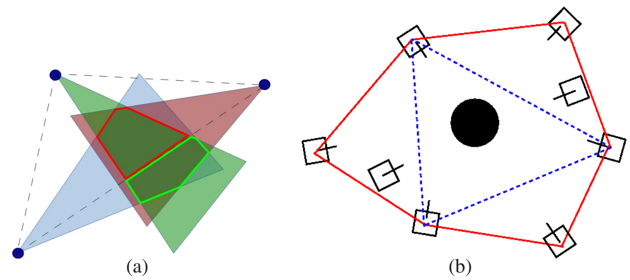


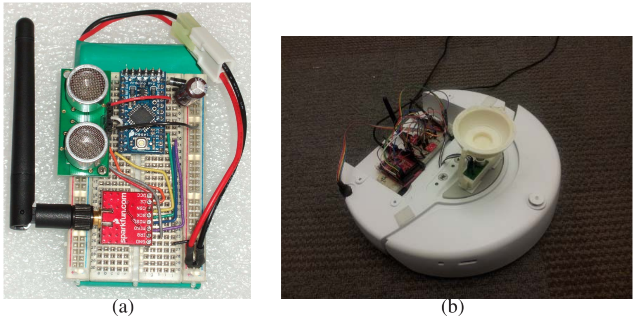
Fig. 1. (a) Example beacon placement and the corresponding coverage classes: blue circles depict beacons; shaded regions depict coverage; the region outlined in red is *well-seen* and the region outlined in green is *well-heard* but not well-seen. (b) Note that a robot (black circle) lies within the convex hull (red lines) of all the beacons (squares) that each currently cover it, if and only if the robot is well-seen (blue dotted lines).

The problem we consider here has relevance for applications in which the robot remains within a specific region (and where other positioning systems such as GPS are impractical), as in *e.g.* the work of Wurman et al. [1]. In such a case, for localization purposes, instrumenting the area with affordable devices might be more desirable than equipping the robot with the sensing capability necessary to perform more general localization using SLAM algorithms as in *e.g.* Montemerlo *et al.* [2].

Consider, for example, an underwater application in which an autonomous underwater vehicle (AUV) is required to periodically measure water parameters such as salinity and temperature from a number of sites located within some region. A network of acoustic beacons might be more affordable (and effective) than other localization options. Such a system can serve multiple robots with no additional overhead. Likewise, an application that automatically transports goods along set routes in an indoor location could benefit from a beacon based positioning system.

In previous work we have looked at localization issues in sensor networks [3] and in 'hybrid' systems that combine both static sensors and a mobile robot [4]. In [4] we used an entropy-based approach for determining a suitably risk-averse set of paths that connect operating points in an environment

978-1-4799-4337-1/14 $31.00 © 2014 IEEE
DOI 10.1109/CRV.2014.28



(a) (b)

Fig. 2. (a) Acoustic ranging beacon and (b) mobile robot used in experimental research.

based on a *fixed* beacon placement. That work assumed that in such systems it is reasonable to expend computational effort initially to later facilitate accurate localization. Here we continue to examine multi-component localization systems: we present a heuristic-based approach for quantifying beacon placements; investigate potential search algorithms that operate over the value returned by our heuristics; evaluate our approach with experiments; and present concluding remarks.

II. Related Work

A number of researchers have considered variants of sensor placement problems for purposes of minimizing uncertainty in robot localization.

The theory related to sensor placement is an active research area. Recently, the generalized sensor placement problem was proved NP-hard by Tekdas and Isler [5]. The authors also presented an approximation algorithm, to compute the minimum number and placement of sensors so that the localization uncertainty at *every* point in the workspace is less than a given threshold. Krause *et al.* [6] demonstrated a constant-factor approximation algorithm to compute the placement and scheduling of sensors simultaneously. In related 'art gallery' problems, such as that considered by Gonzalez-Banos [7], a "valid" beacon placement requires that every point on the floor plan must be covered by at least one beacon. Versions of this problem are also considered by Cary *et al.* [8] and Castro *et al.* [9], [10], in which the beacons are modeled as floodlights with bounded visibility. The problem of localizing a robot in a two dimensional polygon without holes while minimizing the distance traveled by the robot was considered by Rao *et al.* [11].

An interesting subfield is the directional sensor placement problem [12]. Ai and Abouzad proved that maximizing the coverage with directional sensors while minimizing the number of sensors is NP-hard in their work on random directional sensor placement [13]. Recently, Akbarzadeh *et al.* [14] looked at directional sensor placement for optimal coverage using a more real world perspective: where they used a 3D probabilistic sensing model and took into account terrain topography that affects line-of-sight coverage.

Other, less theoretical, approaches depend on metrics for uncertainty. For example, a position error bound was derived by Jourdan and Roy [15] for ultra-wide bandwidth localization in densely cluttered environments. The authors focused on the case where range measurements can be positively biased and their variance depends on the distance between the agent and corresponding beacons. Jourdan *et al.* showed that in dense cluttered environments, non-line-of-sight (NLOS) beacons can convey valuable information, a result that partially contradicts early work on the value of NLOS beacons by Yihong and Kobayashi [16]. In contrast to the work of Jourdan and Roy, our approach trades off an accurate sensor model for computational efficiency. In related work, Martinez and Bullo [17] presented a motion coordination algorithm that steers a mobile sensor network to an optimal deployment in applications that track static and dynamic targets. These authors proved that an "optimal" configuration of the sensors is a placement where the sensors are uniformly located in circular fashion around the target. Recent work [18], [19] attempts to provide a guaranteed bound of the maximum deviation of a robot from its desired path by placing sensors incrementally. Visibility is considered from the robot's perspective.

The problem of minimizing uncertainty in simultaneous localization and mapping was considered by Dudek and colleagues [20], [21] using a vision-based representation of the environment for navigation. Krause *et al.* [22] looked at the problem of optimizing the placement of sensors in a Gaussian process by choosing the placement that is most "informative" about the unsensed locations.

In the next sections we define the specifics of the problem we consider and present the details of our approach.

III. Problem Definition

Given a two dimensional map m of the environment and a specific path p contained within m we wish to select a positioning of N ranging beacons such that a robot attempting to traverse a path p may do so with a high probability of success. We do not as yet seek an "optimal" solution to this problem, which would require a rigorous definition of 'high probability' of success.

Instead, we now present a heuristic-based mechanism for evaluating the quality of a beacon pose (*i.e.* position and orientation) with respect to a particular input path p and then show how we can search over various beacon poses in order to find high quality solutions for path p.

IV. Beacons Placement

A. Overview of Approach

To estimate the utility of a particular beacon placement for a given path p, we make some simplifying assumptions. We assume that the robot can obtain a range estimate from a beacon when the robot is within the beacon's coverage. Furthermore, we assume that near-maximum positional certainty can be obtained when the robot is inside a triangle formed by three beacons, each of which can provide a range estimate to the robot. Based on the beacon locations, we then assign a localization utility to each grid square in the environment. The integral of the localization utility measured along a path p gives the utility of the pose of a set of beacons.

B. Beacons

A *beacon placement* specifies the position and orientation of each of the N deployed beacons $B = \{B_1, B_2, \ldots, B_N\}$ where $B_i = \{x_i, y_i, \theta_i\}$.

A *beacon coverage model* refers to the region, relative to the beacon's pose B_i, in which we may assume the robot can obtain a range estimate from the beacon. In a *homogeneous beacon placement* all beacons use the same coverage model. In the work presented in this paper we use a triangle to describe the coverage model of a beacon. This coverage model is motivated by the physical properties of acoustic beacons.

C. Well-Seen and Well-Heard

We say a region is *well-heard* if each point in the region is covered by three or more beacons. The term "well-heard" is motivated by our use of acoustic beacons in our experimental testbed (see Figure 2). A region is *well-seen* if it is *well-heard* and in addition each point in the region lies within the coverage model of three beacons whose locations determine a triangle containing the point. See Figure 1(a) for an example of *well-heard* and *well-seen* regions.

We use convex hulls to find well-seen regions. For a point x that is covered by three or more beacons, we check whether x is inside the convex hull of the beacon positions. If x is inside the convex hull, then it is well-seen.

Our *well-seen* and *well-heard* heuristics are motivated by trilateration research, such as that by Yang et al. [23] and Dulman et al. [24]. For example, Yang et al. present empirical and simulation results in their Quality of Trilateration (QoT) analysis [23] that support our approach of discretizing trilateration quality based on the geometry of the beacons. The authors compare beacon placements on a circle of fixed radius and present the resulting continuum of trilateration quality; the results can be naturally divided into 'good' and 'bad',w giving support for our heuristic-based classification.

Although calculating the QoT as computed by Yang et al. [23] would arguably serve as a better metric than *well-seen* and *well-heard* because of its finer granularity, it can be computationally intense. To apply QoT to our problem, QoT would need to run on all combinations of line-of-sight beacons at a particular point, which is a combinatorial explosion problem when dealing with large sets of beacons covering the same point. *Well-seen* and *well-heard* do not suffer from the same scaling issue because deciding if a point is *well-seen* is bounded by the running time necessary to build the convex hull for all visible beacons: $O(n \log n)$ [25].

D. Localization Utility

Given a specific beacon placement and the beacon coverage model, any point in the environment can be rapidly classified based on the beacon coverage it receives. Our approach is to map this discrete classification of beacon coverage into a *localization utility*. Table I shows an effective definition of a mapping of beacon coverage to utility.

Coverage	Localization Utility
0 beacons	0
1 beacon	1
2 beacons	2
well-heard	3
well-seen	5

TABLE I
A DEFINITION (USED IN OUR SIMULATIONS) OF *localization utility* AS A FUNCTION OF BEACON COVERAGE.

E. Utility of a Beacon Placement

To compute the utility of a beacon placement B given a path p through a map m we integrate the weighed localization utility along p:

$$f(m, p, B) = \int_p w(x)u(x)dx$$

where $w(x)$ and $u(x)$ give the path importance weight and localization utility, respectively, at point x along the path. Similar assessments of uncertainty metrics along a path have been used before, *e.g.* [26].

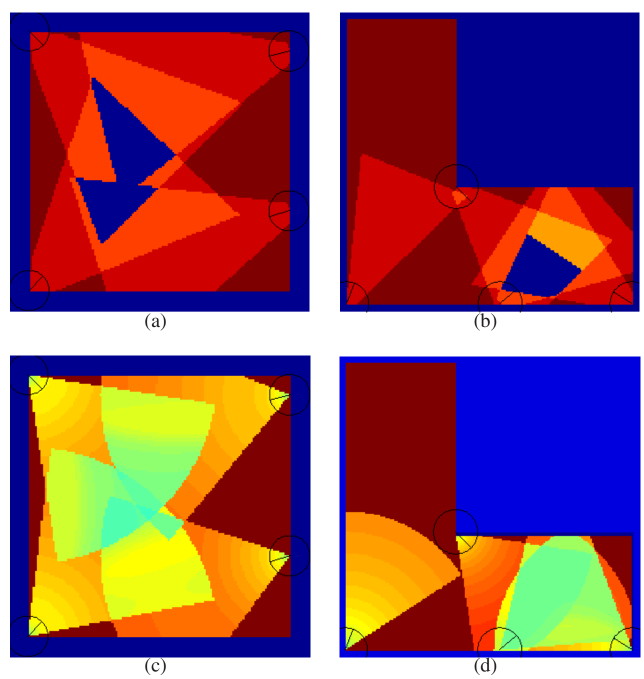


Fig. 3. Example beacon placement and the corresponding localization utilities using our utility-based approach and entropy approach for 'L' shaped and square environments: (a) and (b) utility; (c) and (d) entropy. Circles depict beacon locations and orientations.

1) Search Over Beacon Placements: Using the method above for computing a utility value $u = f(m, p, B)$ for a beacon deployment B for a given path p in environment m, we can search over the space of beacon placements for high quality deployments. We briefly describe several search strategies for evaluating beacon placements.

2) Hill Climbing (HC) : We first define a hill climbing algorithm in which n is the number of beacons, k is the number of beacon poses and f is the function that determines the utility value. We found that 10% of T was a good threshold for when to restart at a new position.

a. Set $t \leftarrow 1$
b. Set $T \leftarrow (k - n) * n$
c. While $t < T$
    1. Select an initial beacon placement B^t randomly
    2. $nc \leftarrow 0$ (nc means "no change")
    3. While $nc < T * 0.10$
        i) Select a single beacon i at random, and select a new random location B_i^* for i.
        ii) Set $B^* \leftarrow \{B_1^t, \ldots, B_i^*, \ldots, B_N^t\}$
        iii) If $f(m, p, B^t) < f(m, p, B^*)$ then
            A) $B^{t+1} \leftarrow B^*$
            B) $nc \leftarrow 0$
        iv) Else
            A) $B^{t+1} \leftarrow B^t$
            B) $nc \leftarrow nc + 1$
        v) $t \leftarrow t + 1$
        vi) If $t >= T$ then *break*

3) Simulated Annealing (SA) : We implemented a simulated annealing variant of the hill climbing approach (HC) described above:

a. Set $\tau \leftarrow 1$
b. Set $t \leftarrow 1$
c. Set $T \leftarrow (k - n) * n$
d. Select an initial beacon placement B^t randomly
e. While $t < T$
    1. Select a single beacon i at random, and select a new random location B_i^* for i.
    2. Set $B^* \leftarrow \{B_1^t, \ldots, B_i^*, \ldots, B_N^t\}$
    3. If $f(m, p, B^t) < f(m, p, B^*)$ then
        i) $B^{t+1} \leftarrow B^*$
        ii) $nc \leftarrow 0$
    4. Else if $\tau > 0$
        i) $\tau \leftarrow exp((f(m, p, B^t) - f(m, p, B^*))/\tau)$
        ii) If $\tau \geq rand()$
            A) $B^{t+1} \leftarrow B^*$
        iii) Else
            A) $B^{t+1} \leftarrow B^t$
    5. Else
        i) $B^{t+1} \leftarrow B^t$
    6. $t \leftarrow t + 1$
    7. $\tau \leftarrow \tau - (1/T)$
    8. If $t >= T$ then *break*

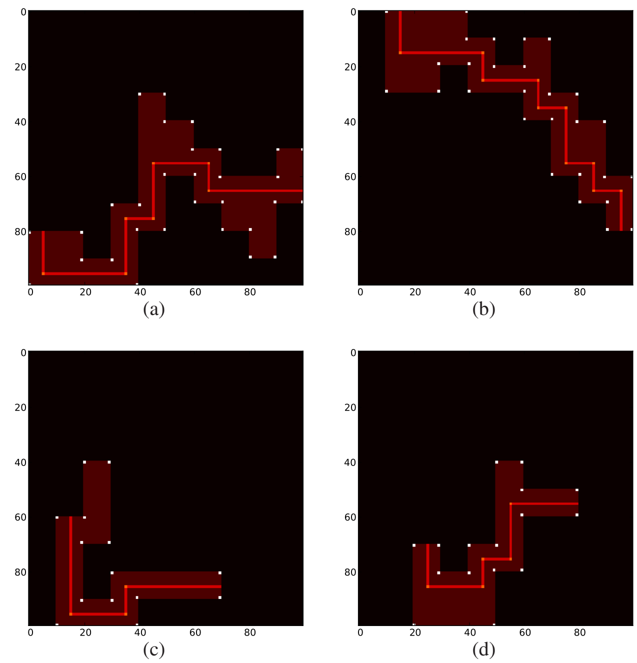


Fig. 4. Examples of random maps used in simulations. Crimson spaces represent navigable areas, orange lines represent the path, yellow squares represent weighted corners along the path and white squares represent possible positions for beacons.

4) Coordinate Descent (CD) : Motivated by the RELOCATE algorithm presented by Jourdan and Roy [26], we implemented a coordinate descent algorithm:

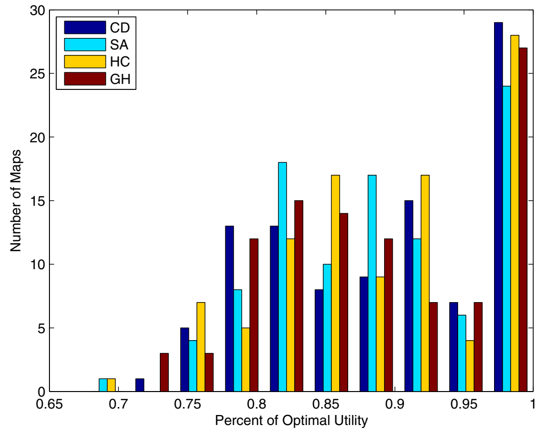
a. Select an initial beacon placement B^0 randomly
b. Select a new placement as follows:
    1. select a single beacon i at random
    2. exhaustively evaluate each new potential location for i to select B_i^* as the locally maximizing B_i^* for $f(m, p, B^*)$ where $B^* \leftarrow \{B_1^t, \ldots, B_i^*, \ldots, B_N^t\}$.
c. set $B^{t+1} \leftarrow B^*$
d. Loop over steps b and c until $t = T$

5) Greedy Heuristic (GH): In order to provide a lower bound for the search algorithms, we implemented a greedy heuristic (GH). GH focuses on moving one beacon at a time to the pose that results in the maximal utility. The algorithm details are as follows:

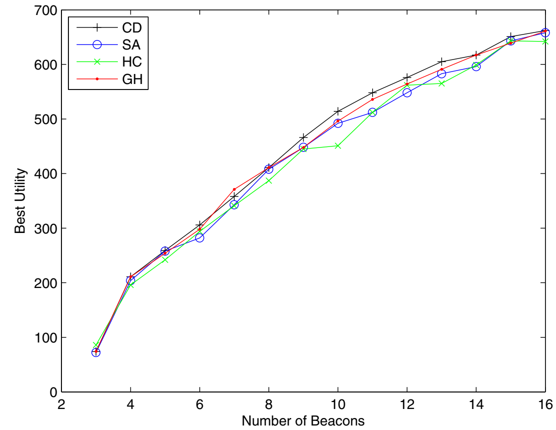
a. Let B be a vector of N uninitialized beacon positions
b. For $i = 1 \rightarrow N$
    1. let $B^* \leftarrow \{B_1, \ldots, B_{i-1}, B_i^*\}$
    2. exhaustively evaluate each new potential location for i to select B_i^* as the locally maximizing B_i^* for $f(m, p, B^*)$.
    3. $B_i \leftarrow B_i^*$

6) Brute Force (BF):

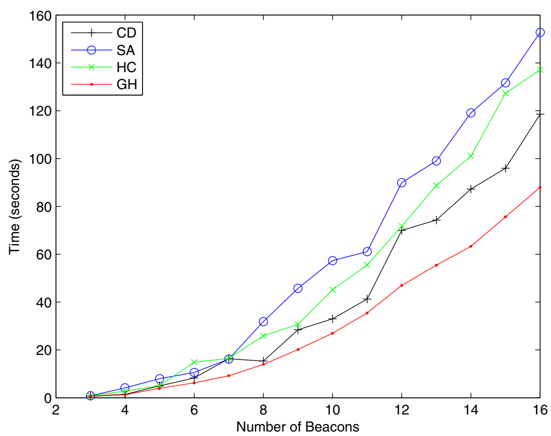
a. Let $M =$ the total number of potential beacon poses.



(a) Histogram showing distribution of utility values as a fraction of optimal on 100 maps with 3 beacons.



(b) Comparison of the utility of search approaches as the number of beacons is increased on the map shown in Figure 4(a).



(c) Comparison of the search algorithms' running times when used to compute the utilities in Figure 5(b).

	Runtime (σ)	% of optimum (σ)
BF	403.085(420.68)	1.000(0.000)
HC	0.344(0.127)	0.895(0.083)
SA	0.403(0.151)	0.890(0.077)
GH	0.408(0.156)	0.895(0.079)
CD	0.295(0.107)	0.888(0.082)

(d) Mean results for the various placement algorithms over 100 maps.

Fig. 5. Results from running the search algorithms on randomly generated maps.

b. Exhaustively search over all $\binom{M}{N}$ possible beacon placements to find the placement that maximizes utility.

V. Results

In our simulations, we restrict the possible orientations for which a beacon can be placed to a discrete set. We considered sixteen orientations: $\theta \in \{0^o, 22.5^o, 45^o, \ldots, 337.5^o\}$. Furthermore, we restrict potential beacon locations to be at vertices at corners on the walls of the map and discard wall facing orientations. Continuous or finer discretizations of potential beacon placements should also work with our approach.

We compare the localization utility computed by our approach throughout a simple environment to a metric calculation of the entropy in the PDF of the robot given realistic beacon models obtained via experimentation (Figure 3). We used the approach described in MacMillan *et al.* [4] for the entropy calculations. It can be seen that our heuristic is effective at identifying areas in the environment in which the robot can be accurately located. Although the entropy based metric is arguably more accurate for assessing the utility of a beacon placement, it is more computationally expensive to obtain; furthermore, the approximations and simplifications required to make the entropy assessment pragmatic affect the accuracy of the result.

In our search approaches, we conducted simulations on various randomly generated maps and paths (see Figure 4 for examples). To generate a random map, we selected a start point and an end point in a region and added a joining hallway using a random walk. We then located potential vertices at each corner of the hallways walls and generated the path by running Dijkstras algorithm from the top leftmost grid point to the top rightmost grid point.

We solved the 3 beacon placement problem on a number of these maps using brute force. We then compared the results

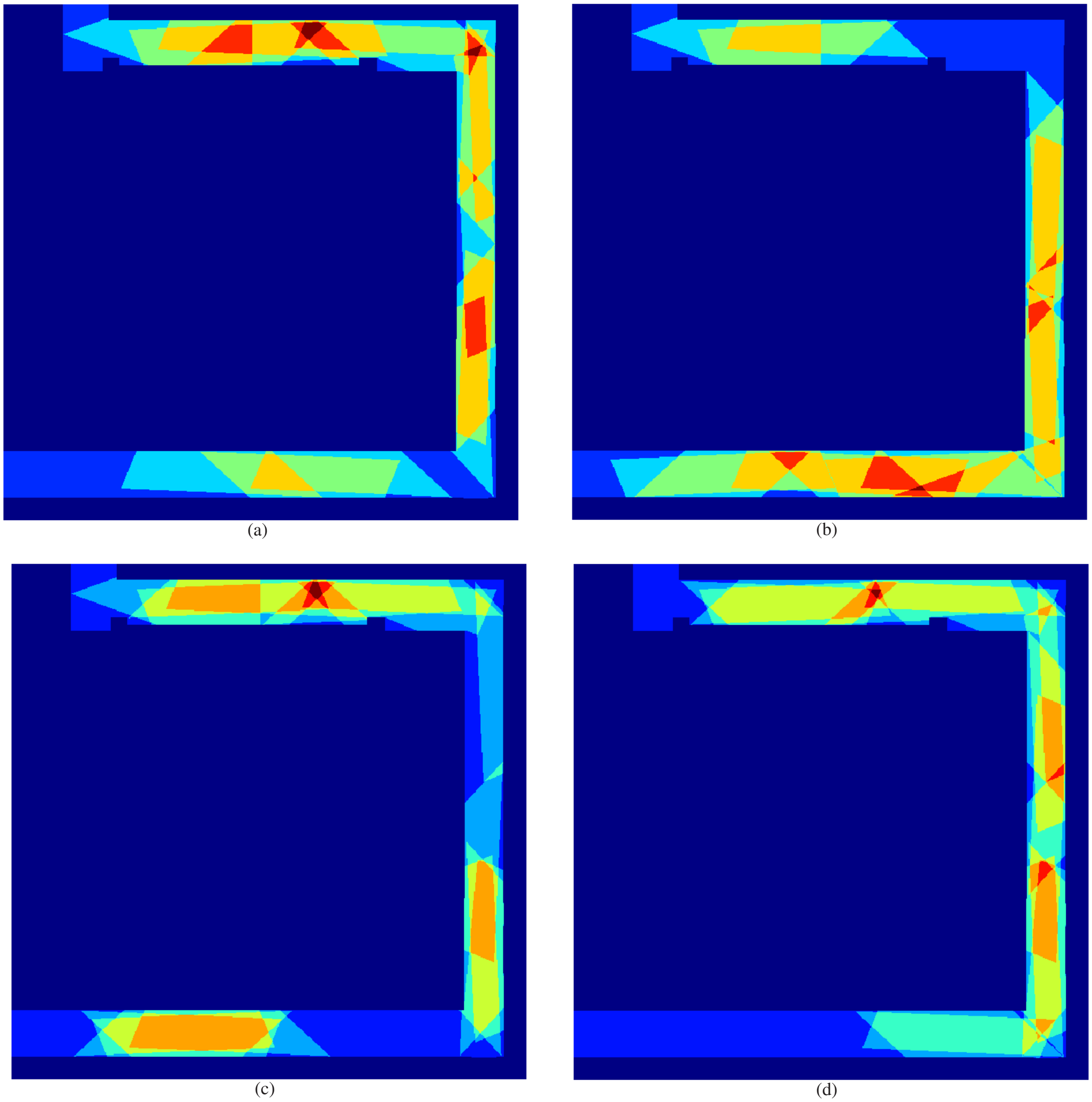


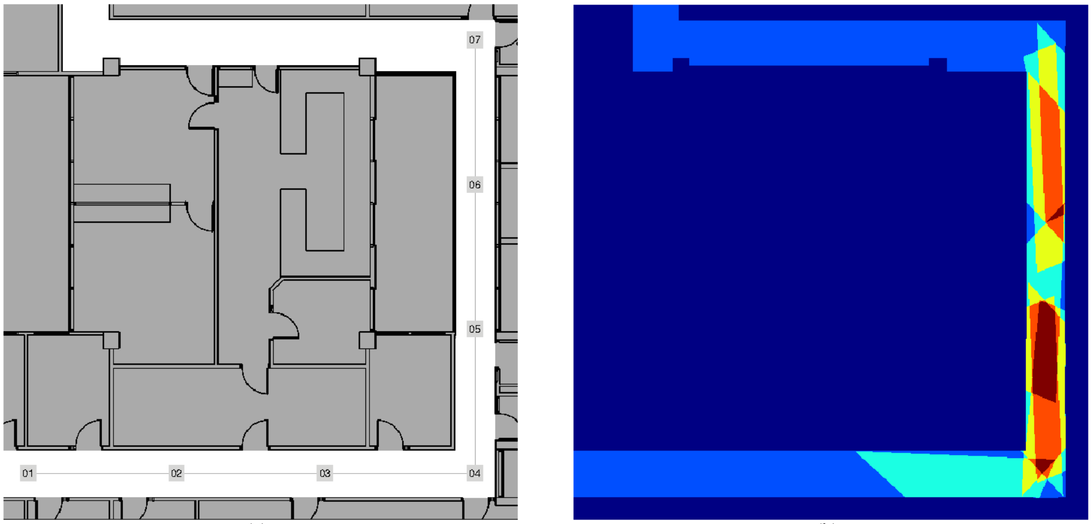
Fig. 6. Placement of 16 beacons on the map (shown in Figure 7(a)) as found by (a) HC, (b) SA, (c) CD, and (d) GH.

from each of the other search approaches to the optimal result obtained via brute force. The brute force approach, however, quickly became intractable as the problem scaled. For example, in an environment of 200 possible beacon poses, the total number of possible poses when given only 10 beacons exceeds 10^{16}.

The beacon placement problem as we have defined it here presents a challenging search problem. Each of the search algorithms considered shows evidence of converging to a local optimum even for simple problem instances.

Figure 5(a) shows a histogram of utility values obtained for the various search algorithms we implemented, on 100 maps. The algorithms produced sub-optimal placements compared to brute force, but performed well considering the number of evaluations performed by the algorithms and their total runtimes as seen in Figure 5(d).

Figure 5(b) demonstrates the effectiveness of the different search algorithms as the number of beacons increases. Note that at this scale, CD, HC and SA perform comparably in terms of utility. GH serves as a good lower bound of performance for the other search algorithms. Although SA tends to perform better than the other search algorithms, it usually takes more time to approach optimality (Figure 5(c)).



(a) (b)

Fig. 7. (a) Map of the floorplan used in our simulations. The white areas represent navigable regions. The black line represents the path and the grey numbered squares on the path are waypoints. (b) Placement of 8 beacons on the map as found by GH.

VI. Experimentation

Our experimentation involved running a robot through an indoor environment instrumented with range-only beacons. We used custom-built beacons comprised of off the shelf components: a Devantech SRF04 ultrasonic rangefinder, Nordic nRF24L01 radio and Arduino Pro Mini (see Figure 2(a)). Our robot consists of an iRobot Create and yaw-gyro controlled by a laptop running the Robot Operating System (ROS). To read distance values from the beacons, the robot also carries a radio, ultrasonic rangefinder and Seeeduino Mega (see Figure 2(b)). Time-difference-of-arrival between a beacon's ultrasonic pulse and a radio transmission determines the beacon-robot distance. To test the effectiveness of our method, we ran the robot through a section of the fourth floor of the Engineering/Computer Science (ECS) building at the University of Victoria, Canada (see Figure 7(a)).

While we had only 8 beacons with which to run physical experiments, it was feasible to run the simulation with more beacons. In fact, we ran the simulations with 16 beacons and recorded the results for each algorithm (see Figure 6). We programmed the robot to travel one meter as measured by its internal odometry and then to pause. At each such observational pause the robot then pinged the nearby beacons, using its knowledge of the beacon deployment B. Then the robot would update its particle filter with its distance observations to the beacons. We also programmed the robot to pause when it perceived itself to be at each waypoint. At such waypoint pauses, we measured by hand the distance between the robot and the true location of the corresponding waypoint on the path. We call this distance the *error* for that waypoint. These error measurements indicate the deviation of the robot's path from the desired path.

Figure 8 shows the error at each waypoint for three separate runs of the robot. Observe that the error is the highest at waypoint 3. This may be explained as follows. None of the programmed observational pauses preceding the third waypoint could occur at a well-seen point. Thus errors accumulated. The error is minimum at waypoint 4, which may be explained as follows. For each of the runs, we observed that the robot's last observational pause, before the waypoint 4 pause, occurred at a well-seen point.

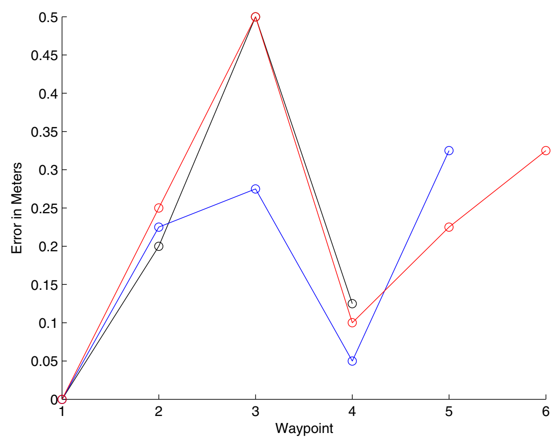


Fig. 8. **Error** at each waypoint, i.e., the distance in meters between the true waypoint i on the path and the position at which the robot stops when it believes it is at waypoint i.

VII. Discussion

We have proposed a novel, computational geometry-based approach to solving the problem of ranging beacon placement for the navigation of a mobile robot. We have shown that our heuristic based approach allows a computationally inexpensive assessment of beacon placements. By discretization of the environment through triangulation, we have shown that standard search approaches can find high quality placements.

In future work we will look further at incorporating the dead reckoning capabilities of the robot into the path assessment portion of the beacon assessment algorithm. Specifically, we plan to apply a motion model for a robot and estimate positional uncertainty in the temporal domain. In addition, we plan to analyze more sophisticated coverage functions that provide greater granularity metrics (*e.g.* QoT, noise-resistant metrics, fat triangles) for beacon placements at the cost of greater computational complexity. Finally, we will investigate scaling our approach to large environments that we decompose into polygonal regions. Good beacon placements can then be found for each polygon region independently.

Acknowledgments

We thank Sean Chester and Laurie Heyer for stimulating discussions and Mantis Cheng for technical assistance and lab space. We acknowledge the Natural Sciences and Engineering Research Council of Canada for their funding.

References

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," in *AAAI'07: Proc. of the 22nd National Conf. on Artificial Intelligence*. Vancouver, British Columbia, Canada: AAAI Press, 2007, pp. 1752–1759.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. of the Sixteenth Int. Joint Conf. on Artificial Intelligence (IJCAI)*. Acapulco, Mexico: IJCAI, 2003.

[3] D. Marinakis, N. MacMillan, R. Allen, and S. Whitesides, "Simultaneous localization and environmental mapping with a sensor network," in *Proc. of ICRA 2011*, Shanghai, China, May 2011, pp. 5881–5886.

[4] N. MacMillan, R. Allen, D. Marinakis, and S. Whitesides, "Range-based navigation system for a mobile robot," in *CRV 2011*, St. John's, Canada, May 2011, pp. 16–23.

[5] O. Tekdas and V. Isler, "Sensor placement for triangulation-based localization," in *IEEE Trans. on Automation Science and Engineering*, July 2010, pp. 681–685.

[6] A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin, "Simultaneous placement and scheduling of sensors," in *Proc. of the 2009 Int. Conf. on Information Processing in Sensor Networks*, ser. IPSN '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 181–192.

[7] H. González-Banos, "A randomized art-gallery algorithm for sensor placement," in *17th Annual Symposium on Computational Geometry*, ser. SCG '01. New York, USA: ACM, 2001, pp. 232–240.

[8] M. Cary, A. Rudra, A. Sabharwal, and E. Vee, "Floodlight illumination of infinite wedges," *Computational Geometry Theory and Applications*, vol. 43, pp. 23–34, January 2010.

[9] V. Estivill-Castro and J. Urrutia, "Optimal floodlight illumination of orthogonal art galleries," in *CCCG*, 1994, pp. 81–86.

[10] ——, "Two-floodlight illumination of convex polygons," in *4th Int. Workshop on Algorithms and Data Structures*, ser. WADS '95. London, UK: Springer-Verlag, 1995, pp. 62–73.

[11] M. Rao, G. Dudek, and S. Whitesides, "Randomized algorithms for minimum distance localization," *Int. J. of Robotic Research*, vol. 26, no. 9, pp. 917–933, 2007.

[12] M. Amac Guvensan and A. Gokhan Yavuz, "On coverage issues in directional sensor networks: A survey," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1238–1255, September 2011.

[13] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 1, pp. 21–41, February 2006.

[14] V. Akbarzadeh, C. Gagne, M. Parizeau, M. Argany, and M. Mostafavi, "Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 2, pp. 293–303, 2013.

[15] D. B. Jourdan, D. Dardari, and M. Z. Win, "Position error bound for UWB localization in dense cluttered environments," in *IEEE Int. Conf. on Communications (ICC06)*, June 2006, pp. 3705–3710.

[16] Y. Qi and H. Kobayashi, "Cramér-rao lower bound for geolocation in non-line-of-sight environment," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2002)*, 2002, pp. 2473–2476.

[17] S. Martinez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[18] M. P. Vitus and C. J. Tomlin, "Sensor placement for improved robotic navigation," in *Robotics: Science and Systems*, Zaragoza, Spain, 2010.

[19] M. Beinhofer, J. Mller, and W. Burgard, "Effective landmark placement for accurate and reliable mobile robot navigation," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1060–1069, October 2013.

[20] R. Sim, G. Dudek, and N. Roy, "Online control policy optimization for minimizing map uncertainty during exploration," in *IEEE Int. Conf. on Robotics & Automation (ICRA 2004)*, 2004, pp. 1758–1763.

[21] D. Meger, I. Rekleitis, and G. Dudek, "Heuristic search planning to reduce exploration uncertainty," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2008)*, September 2008, pp. 3392–3399.

[22] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *J. of Machine Learning Research*, vol. 9, pp. 235–284, June 2008.

[23] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 99, pp. 631–640, 2010.

[24] S. O. Dulman, A. Baggio, P. J. M. Havinga, and K. G. Langendoen, "A geometrical perspective on localization," in *Proc. of the first ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*. New York: ACM, September 2008, pp. 85–90.

[25] T. M. Chan, "Optimal output-sensitive convex hull algorithms in two and three dimensions," *Discrete & Computational Geometry*, vol. 16, no. 4, pp. 361–368, 1996.

[26] D. B. Jourdan and N. Roy, "Optimal sensor placement for agent localization," *ACM Trans. on Sensor Networks*, vol. 4, no. 3, pp. 1–40, May 2008.