Active Sensing Data Collection with Autonomous Mobile Robots

Richard Wang, Manuela Veloso, and Srinivasan Seshan
Carnegie Mellon University

***Abstract*— With the introduction of autonomous robots that help perform various tasks in our environments, we can opportunistically use them for collecting fine-grain sensor measurements about our surroundings. Use of mobile robots for data collection scales much better than static sensors in terms of number of measurement locations and provide more fine-grain accuracy and reliability than alternate human crowd-sourcing efforts. One of the unique features of mobile robots is the ability to control and direct where and when measurements should be collected. In this paper, we present a system to compute paths for the robot to follow that incorporates the robot's limited expected deployment time, expected measurement value at each location, and a history of when each location was last visited.**

I. Introduction

It is difficult for traditional data collection efforts to collect accurate, fine-grain, and reliable sensor measurements across large, indoor environments. Static sensors do not scale well as the hardware requirements increase quickly with large environments. Even dense static sensor deployments cannot capture fine-grain measurements so they often use interpolation techniques to estimate fine-grain variations [1], [2]. Crowd-sourced data collection from humans with mobile phones has been popular but it remains a challenge to accurately localize these devices and deal with sensor variations across different devices [3], [4], [5].

We can use autonomous robots to complement these alternate measurement collection efforts. The major advantage of mobile robots as data collection devices is continuous localization with high accuracy, automatic navigation to numerous locations, and reliable measurement collection under similar mounting conditions [6], [7], [8]. Each robot only requires mounting one sensing device so it is relatively low cost to collect measurements across large environments. In addition, the ability to dynamically adjust data collection strategies provides great flexibility. The challenge of using mobile robots is their limited battery life and uncertain operating time.

In this work, we investigate the challenge of deciding where the robot should direct its active data collection efforts in indoor environments when faced with uncertainties from interruptions from higher priority tasks. Some measurement locations may have higher value than others and therefore deserve more frequent visits. As a result, an intelligent collection strategy needs to balance the value of measurements at various locations with the navigation time between locations as well as the robot's expected deployment time.

We focus on creating a flexible framework that adjusts navigation strategies across multiple deployments. Our approach is centered around maintaining a history of data collection efforts over time and formulating the problem as a reward collecting traveling salesman problem (TSP). We evaluate our framework under several variations of the traveling salesman problem and find the discounted-reward TSP performs well when faced with operation uncertainties.

II. Background

We introduce background on autonomous robots and various sensors. We then discuss how easy it is to equip autonomous robots for sensor data measurement collection.

A. Autonomous Robots

Mobile robots move around an indoor environment on their own without human intervention. As shown in Figure 1, the robot used in this study has dimensions (1.2m x 0.5m x 0.5m) with an omni-directional wheeled based for driving across the environment. An on-board tablet performs all robot computations and also provides a GUI for interacting with humans. The robot is equipped with perceptive sensors including LIDAR and Kinect that can provide centimeter-level localization accuracy and a battery that sustains up to four hours of continuous navigation on a single charge. We will define robot location updates as a time and position tuple $\langle t_i, x_i, y_i \rangle$

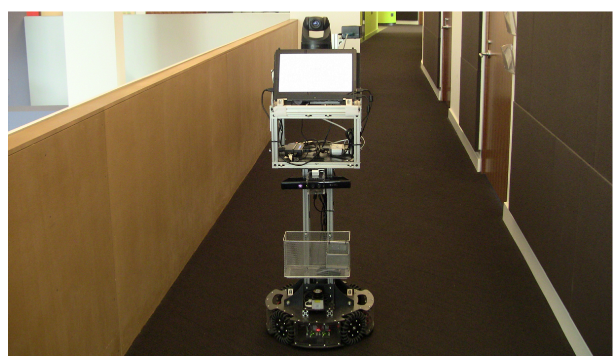


Fig. 1: An autonomous robot equipped for data collection.

B. Sensors

It is relatively easy to mount additional sensors on to a robot when compared with smaller more mobile devices like cell phones. A large platform enables one to mount various sensors including indoor climate (temperature and humidity), air quality (CO, CO2, Nitrogen, etc.), and wireless signals. Each measurement update can be represented as a time and value tuple $\langle t_i, s_i \rangle$. These types of measurements can be used for a variety of purposes including building management [9], [10] and wireless infrastructure management [11].

C. Sensor Data Collection with Autonomous Robots

While in operation, the robot can easily record sensor measurements as well as the corresponding time and location as it moves across the environment. Given that different sensors update at different rates, one can align the measurement updates $\langle t_i, s_i \rangle$ by timestamp to identify the best location estimate $\langle t_i, x_i, y_i \rangle$ for each sensor measurement $\langle t_i, x_i, y_i, s_i \rangle$. Sensors that update at a low frequency (like indoor climate sensors for temperature and humidity) may result in the robot moving several meters between successive sensor updates when operating at normal speeds, making an accurate location estimate less certain. As a result, it may sometimes be necessary for the robot to move more slowly when collecting certain types of sensor measurements.

In general, there are two robot deployment modes. The first is when the robot is performing some task so we can only opportunistically collect sensor data wherever the robot happens to navigate through. The second is when the robot has no tasks to perform so it can actively decide where to focus its data collection efforts. This second use case is the focus of this paper.

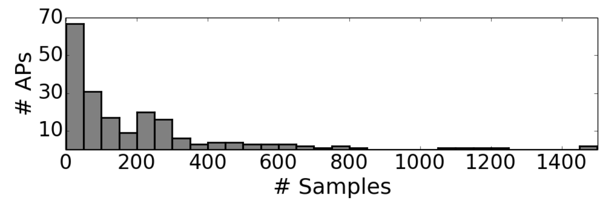
III. Data, Data, Everywhere But Here

We first motivate the need for autonomous robots to proactively navigate by showing the unique challenges of data collected by a mobile platform. We show data collected from real-world deployments of autonomous robots in an enterprise environment to show that while the aggregate volume of data and number of unique locations visited is large, the number of measurements at any specific location is small. This is a unique challenge when compared to a static sensor that provides fine-grain temporal visibility but only from a single location. When coupled with limited deployment times, there is a clear need for robots to be intelligent about where they decide to collect measurements to ensure up-to-date and fine-grain sensor maps.

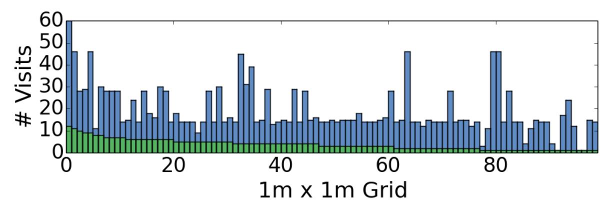
As an example, we collect 802.11 WiFi measurements in this work because there are many possible sensor configurations and the sensor measurements vary considerably over just a few meters, which makes them difficult to predict. Variations in the wireless signals are highly dependent on the transmitting device. We collected measurements with an autonomous robot deployed over a 30 day period. These measurements targeted wireless measurements from static infrastructure access points (APs).

Figure 2a shows how the number of wireless measurements differs across all observed APs. Some APs are measured frequently - for example, one AP is measured over 1400 times. However, many APs are observed less than 50 times. Part of the reason is that each AP broadcasts packets every 100 ms. APs that are communicating with nearby wireless devices provide additional wireless measurement samples, which helps to explain why some APs have many more measurement samples.

Even for APs with a large number of measurements, further segmentation based on factors like location reveals the limited number of measurements for any specific location. Figure 2b shows one AP which had 542 wireless samples. We segmented the measurements by 1m x 1m grid areas. From this, we can see variations across the number of visits for each unique grid. In addition, we also see that just because the robot visited a location does not guarantee that it collected any measurements for this specific AP. Despite a large number of wireless measurements collected, we can see that the view of an AP at any particular location varies a lot, which supports the need for active data collection.



(a) Histogram comparing number of wireless measurements collected for each AP.



(b) # unique visits with measurements for each grid (green) and # total visits per grid (blue, unstacked)

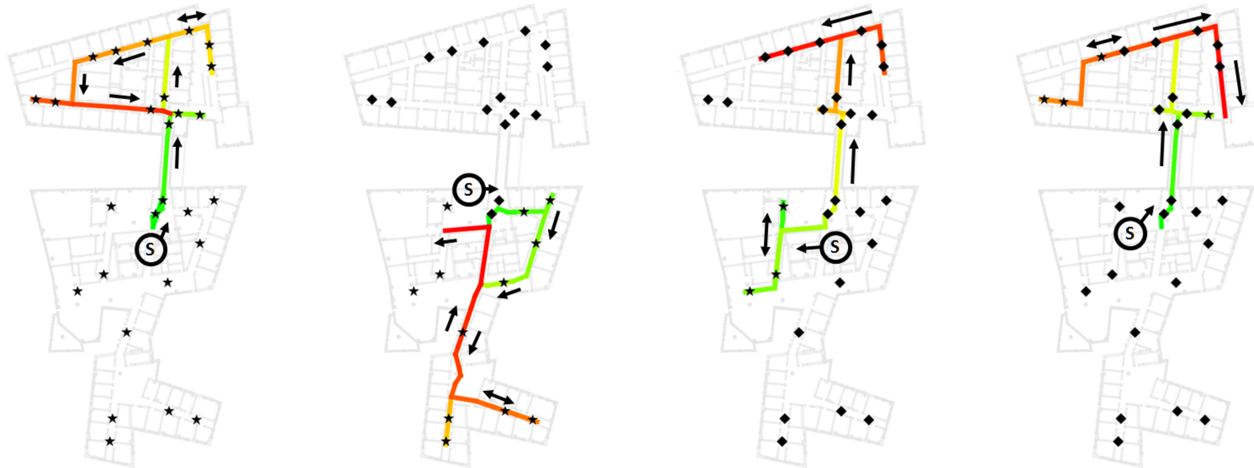
Fig. 2: Showing various ways in which we can refine analysis of various sensor specific features that can lead to sparse measurements for certain sensor configurations.

IV. Path Planning for Active Measurement Collection

By tracking where the robot previously collected sensor measurements, the robot over the course of multiple deployments should intelligently navigate to ensure best effort coverage and periodically revisit important measurement locations over time.

A. Path Planning for Data Collection

Our approach computes navigation paths at the beginning of each active data collection deployment by combining the robot's starting location, expected deployment time, and the measurement value of different locations. To contend with the numerous factors that discriminate navigation path for collecting measurements, we divide relevant factors into either costs or one-time rewards for each measurement location. Costs include navigation and measurement time while rewards estimate the expected measurement value. Rewards are readjusted between deployments so that recently visited locations have lower priority than other less recently visited ones. Therefore, two successive deployments from the same starting location should result in very different navigation strategies.



(a) Deployment #1 (b) Deployment #2 (c) Deployment #3 (d) Deployment #4

Fig. 3: Different navigation path choices across multiple deployments. High priority locations (triangle) have not been visited in the previous three deployments versus low priority locations (square) have been recently visited. Start position indicated with the circle S. The navigation order is indicated by the gradient from green to red as well as select arrows.

One possible representation of the environment would be an occupancy grid; however, our intent is not for the robot to stop frequently at each grid location as it creates inconvenient obstacles for humans moving around. Instead, our intent is for the robot to continuously move while collecting measurements with some flexibility in adjusting the robot's speed based on measurement needs. Therefore, measurement locations are identified via segments of edges e_{meas} along the robot's original navigation graph. The paths computed identify the order in which the robot should visit these measurement edges e_{meas}. This computation needs to balance the costs and rewards as well as the uncertainty about the robot's actual deployment time.

B. Data Collection System

The data collection system is composed of three main pieces: contextual data records, path planning computation, and robot deployment. As shown in Figure 4, data flow across these pieces is important for ensuring the robot executes intelligent data collection strategies over time.

Contextual data records include the robot's original navigation graph and sensor measurement cost-reward parameters for each measurement edge $\langle e_{meas}, c_{meas}, r_{meas} \rangle$. With up-to-date contextual data, the robot can compute a navigation path $\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle ... \in P$ at the beginning of each deployment starting from its current location $\langle x_{init}, y_{init} \rangle$. The robot executes the path while it is deployed, continuously updating the contextual data records as it collects measurements at different measurement locations e_{meas} in the environment. Even if the robot is unable to complete the entire path traversal, the data records are updated to re-weight measurement priorities so that less frequently visited locations will be more likely to be visited across multiple deployments.

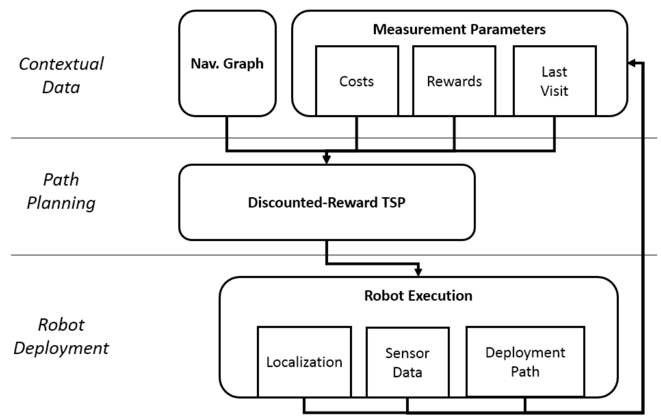


Fig. 4: Data flow across the major pieces of our active data collection system.

C. Formulation as Prize Collecting TSP

We assume that the robot wants to visit all measurement locations if time permits. With uncertain deployment times, the robot needs to carefully consider the order in which it visits each measurement location e_{meas}. In addition, the robot's underlying navigation graph $\langle V, E \rangle \in G$ constrains how long it takes to navigate between different measurement locations. Attached to each measurement location are cost and reward parameters $\langle e_i, c_i, r_i \rangle$, which allows us to compute navigation paths using variants of the prize collecting traveling salesman problem. Reward adjustments between deployments will incentivize measurement collection from less visited locations in subsequent deployments.

1) Cost-Reward Parameters: The robot's active data collection strategy can be influenced by adjusting the cost and reward parameters.

There are two types of costs for each measurement location: navigation cost c_{nav} and measurement cost c_{meas}. The navigation cost is based on traversal time, the time to cover the shortest distance across the edge at normal speeds. Measurement cost can be much higher for reasons like slower speeds for more accurate sensor measurement locations. As a result, the total cost c_i of an measurement location is either its navigation cost or measurement cost $c_i = \{c_{nav} | c_{meas}\}$ depending on whether the robot is simply driving through or capturing measurements along the edge.

Rewards discriminate the relative importance of different measurement locations. The robot collects one-time rewards for each measurement location r_i for each deployment that are either: no reward r_{no}, low priority reward r_{low}, or high priority reward r_{high}. The total reward is simply the sum of these rewards $r_i = r_{no} + r_{low} + r_{high}$. As an example, the low priority rewards can be the difference in time from the current time and time since last visit to help bias unvisited locations. The high priority rewards provide additional flexibility to further influence the robot's navigation for particularly important measurement locations. For example, if we wish for a location to be visited once a week, then we adjust r_{high} based on this condition.

2) TSP for Computing Paths: With the formulation of each measurement location as $\langle e_i, c_i, r_i \rangle$, we can construct a tour to visit all locations and compute a navigation path using variants of the prize collecting traveling salesman problem. Notice that the traditional traveling salesman problem solely considers finding the shortest path cycle and does not consider measurement value.

$$min \sum_{0}^{\infty} c_i \tag{1}$$

In contrast, the prize collecting TSP also consider rewards collected. One variant is called the orienteering problem, which finds the maximum reward path within a fixed time limit MAX_COST.

$$\begin{aligned} max \sum_{0}^{j} r_i \\ s.t. \sum_{0}^{j} c_i \leq MAX\_COST \end{aligned} \tag{2}$$

Another variant called the discounted-reward TSP (DRTSP) finds the maximum discounted reward path with discount rate γ. The discounted-reward results in paths biased towards earlier than later collection of rewards.

$$max \sum_{0}^{\infty} \frac{r_i}{(1+\gamma)^{c_i}} \tag{3}$$

Equation 4 shows how one can select an appropriate discount rate γ by providing the proportion of reward p that should remain after some expected deployment time t. For example, in our evaluation, we aim to discount the reward by 50% of the original value by the time 20 minutes have passed. As a result, we use a discount rate of 1.2%. By adjusting the discount rate, one can easily influence the importance of collecting measurements earlier or later during a deployment.

$$\begin{aligned} \frac{1}{(1+\gamma)^t} = p \\ \gamma = e^{\frac{ln(\frac{1}{p}))}{t}} - 1 \end{aligned} \tag{4}$$

Solving these variants of the TSP is NP-hard but we are working with metric maps, which is the special case Euclidean TSP. As a result, we can use the minimum spanning tree heuristic in combination with the triangle inequality to find paths within a factor of two optimal. Unfortunately, this guarantee no longer holds if the measurement cost is not the same as the navigation cost. Given that the robot computes a different path for each deployment, we continued using the heuristic because it is more worthwhile for the robot to continuously move than wait for an optimal solution.

3) Adjusting Rewards Across Deployments: After each deployment, readjusting the rewards for recently visited measurement locations will ensure variability in the navigation paths for subsequent deployments. There are many potential strategies to readjust these weights. In our evaluation, we consider the simple goal of ensuring each measurement location is visited once every three deployments. Therefore, recently visited measurement locations are assigned a much lower reward value. We will show that these efforts result in intelligent navigation strategies that emphasize data collection.

V. Evaluation

In our evaluation, we wish to show that a robot can be intelligent about active data collection under uncertain deployment times. We first compare how different TSP variants balance the tradeoffs between cost and rewards during a single deployment. We then illustrate how the robot follows different navigation paths across deployments. Finally, we show how these strategies perform under long-term deployments, where the robot is periodically deployed under variable deployment times.

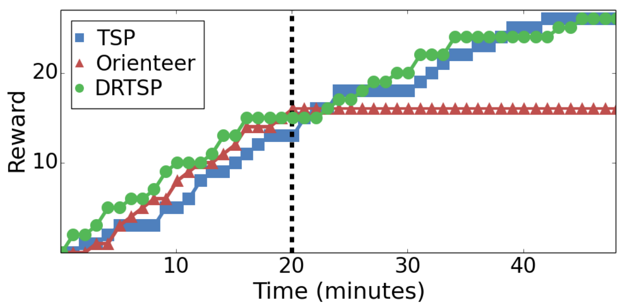
A. Single Deployment Path Strategies

We first look at how much reward would be collected if the deployment was interrupted at any point in time. We consider a robot with an expected deployment time of 20 minutes. From a real navigation map of an enterprise environment, we have marked 25 high priority hallway segments assigned uniform expected measurement reward value of 1. The remaining hallways can be optionally visited with a reward of .01 to incentivize traveling through different hallways. The orienteering algorithm is provided a 20 minute limit. The DRTSP uses a discount rate of 1.2%.

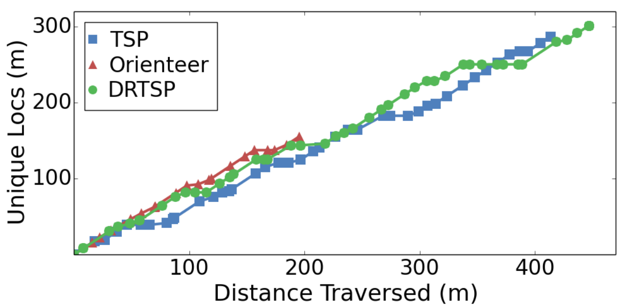
The accumulated rewards from one deployment are shown in Figure 5a while the total distance traveled versus number of unique locations traversed is shown in Figure 5b. There are clear differences across the three path planning algorithms.

First, the original TSP naturally finds a path that takes the least amount of time since it only consider the costs. Both the orienteering problem and discounted-reward TSP (DRTSP) also consider the reward collected so it is no surprise that they find paths with higher initial rewards.

The orienteering solution excels when the exact deployment time is known but has no contingency plan should there be additional time. In contrast, the DRTSP has a plan for additional deployment time making it well equipped when there is uncertainty regarding the robot's deployment time. Due to the small .01 reward for visiting different locations, DRTSP traverses an additional unique 14.16 meters in exchange for traveling 33.22 meters more than the shortest cycle path discovered by the TSP.



(a) Reward collected over time for one path.



(b) Total distance traveled versus # unique locations visited

Fig. 5: Comparing differences in rewards collected by path planning algorithms. Dashed line in Figure 5a indicates the target 20 minute time limit. Higher up-front rewards are desirable when faced with limited collection time.

B. Across Multiple Deployments

We now show how navigation paths evolve over multiple deployments due to simple readjustments of rewards after each deployment. We apply a simply strategy that assigns a reward of 1 for measurement locations not visited in the last three deployments and 0.1 for those that have. Figure 3 shows how this simple adjustment of rewards influences the path traversed when the robot is given a fixed 20 minute time limit each deployment and starts from the same initial position. It takes three deployments to fully cover the environment at which point the navigation strategy focuses on visiting low priority locations. As we can see, adjusting the rewards provides flexibility in deciding where the robot should emphasize its data collection efforts.

Next, we wish to see how effective these navigation paths are able to cover measurement locations over multiple deployments under uncertain deployment times. We consider deployment times randomly generated from Gaussian distribution with mean of 20 minutes and standard deviation of 10 minutes all starting from the same initial location. Rewards are adjusted after each deployment as before. Orienteering is once again given a 20 minute time limit and DRTSP is given a 1.2% discount rate. Figure 6 shows the proportion of measurement locations visited within the last three deployments. It is desirable to have higher proportion of recently visited measurement locations.

Since the TSP does not consider rewards, it simply executes the same navigation path and serves as a baseline. The orienteering solution is able to maintain higher proportion of recently visited locations but it fails to take full advantage of situations where the robot has additional time. The DRTSP is most effective at ensuring it maintain fresh measurements since it emphasizes paths that collect rewards early while also having a plan in case the robot is deployed longer than the expected deployment time.

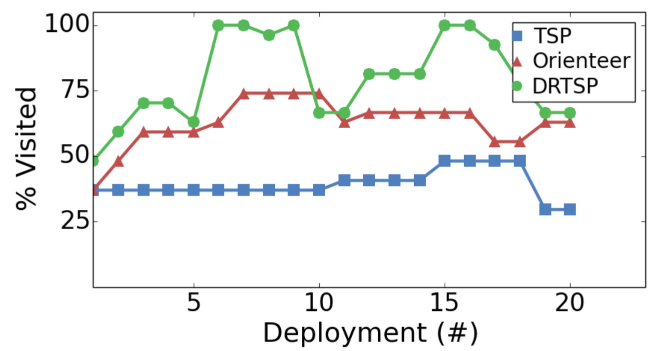


Fig. 6: Percentage of measurement locations visited in the previous three deployments.

VI. Related Work

Related efforts for collecting measurements have different limitations from robots, which makes use of autonomous robots a complementary effort. Some alternate efforts that use mobile robots do not focus on robots that opportunistically collect sensor measurements.

Dense static sensor deployments enable simultaneous data collection at numerous locations but require a significant amount of hardware, must contend with limited physical placement options and still requires human effort to mark sensor locations [12], [11]. High cost and initial labor costs often means that it is difficult to upgrade system hardware. In general, it is difficult to capture fine-grain measurements that a robot can so static sensor efforts use geo-statistical interpolation efforts like Kriging [1] and Gaussian Processes [2] that rely on assumptions about the co-variance of nearby measurement locations. Of course, static sensors do not move so they cannot even consider active path planning issues.

Manual efforts, such as wardriving or site surveys, require people to walk around the building with some sensing device. Users periodically trigger measurement recordings by stopping and then marking their location on a given map [13], [14]. Unfortunately, collected sensor data is susceptible to human influences. More recent efforts automatically estimate device location [15], [3], [4], [5] but they have much worse accuracy than mobile robots equipped with much more powerful extrinsic sensors [6], [7].

Our work differs from the use of mobile robots for coverage and exploration. Coverage focuses on the distributed problem of ensuring multiple robots maintain connectivity [16], [17] while exploration tackles the dynamic issue of exploring a new environment [18].

Mobile robots have also been used as data mules for collecting data wirelessly from static sensors dispersed across an environment for both wheeled [19], [20] and underwater [21] robots. Some such efforts focus on optimizing the paths traversed, which tend to focus on the TSP that emphasizes neighborhoods [22]. Recent work also targeting data collection emphasizes minimizing the combined movement and transmission costs [23], [24]. Across most of these works, the paths computed are intended to be repeated periodically by the robot without interruption, which differs from our goal of generating trajectories for robots with limited and uncertain data collection time.

VII. Conclusion and Future Work

We contribute a framework for enabling autonomous robots to actively navigate across the environment for the purposes of data collection when it has free time. There is often uncertainty regarding how long these opportunities will be so we investigated navigation strategies that adjust based on data collected from previous deployments. We showed this problem can be formulated at the prize collecting traveling salesman problem and that the discounted-reward TSP is well-suited for taking advantage of uncertain deployment times.

For future work, there are opportunities to further extend the algorithm for sensor hardware that require adjustment of robot speeds for data collection. For example, indoor climate sensors that capture temperature and humidity take some time for measurements values to reach equilibrium. There are also opportunities to generate active data collection trajectories that combine potentially conflicting measurement needs of multiple sensors.

References

[1] C. Phillips, M. Ton, D. Sicker, and D. Grunwald, "Practical radio environment mapping with geostatistics," in *Dynamic Spectrum Access Networks (DYSPAN), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 422–433.

[2] B. Ferris, D. Fox, and N. D. Lawrence, "Wifi-slam using gaussian process latent variable models." in *IJCAI*, vol. 7, 2007, pp. 2480–2485.

[3] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "Spot localization using phy layer information," in *Proceedings of ACM MOBISYS*, 2012.

[4] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "Fm-based indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 169–182.

[5] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 197–210.

[6] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," *AAAI/IAAI*, vol. 1999, pp. 343–349, 1999.

[7] J. Biswas and M. Veloso, "Multi-sensor mobile robot localization for diverse environments," *RoboCup 2013: Robot Soccer World Cup XVII*, 2013.

[8] R. Fish, M. Flickinger, and J. Lepreau, "Mobile emulab: A robotic wireless and sensor network testbed," in *IEEE INFOCOM*, 2006.

[9] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*. ACM, 2010, pp. 1–6.

[10] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[11] P. Bahl, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Dair: A framework for managing enterprise wireless networks using desktop infrastructure," in *HotNets IV*, 2005.

[12] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Analyzing the mac-level behavior of wireless networks in the wild," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 75–86, 2006.

[13] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.

[14] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 205–218.

[15] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 293–304.

[16] J. Reich and E. Sklar, "Robot-sensor networks for search and rescue," in *IEEE International Workshop on Safety, Security and Rescue Robotics*, 2006.

[17] Y. Wang and C.-H. Wu, "Robot-assisted sensor network deployment and data collection," in *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*. IEEE, 2007, pp. 467–472.

[18] M. Batalin, G. S. Sukhatme *et al.*, "The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment," *Robotics, IEEE Transactions on*, vol. 23, no. 4, pp. 661–675, 2007.

[19] Y. Tirta, Z. Li, Y.-H. Lu, and S. Bagchi, "Efficient collection of sensor data in remote fields using mobile collectors," in *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*. IEEE, 2004, pp. 515–519.

[20] O. Tekdas, V. Isler, J. H. Lim, and A. Terzis, "Using mobile robots to harvest data from sensor fields," *IEEE Wireless Communications*, vol. 16, no. 1, p. 22, 2009.

[21] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, "Data muling over underwater wireless sensor networks using an autonomous underwater vehicle," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2091–2098.

[22] B. Yuan, M. Orlowska, and S. Sadiq, "On the optimal robot routing problem in wireless sensor networks," *IEEE Transactions on Knowledge & Data Engineering*, no. 9, pp. 1252–1261, 2007.

[23] J. Goerner, N. Chakraborty, and K. Sycara, "Energy efficient data collection with mobile robots in heterogeneous sensor networks," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2527–2533.

[24] D. Ciullo, G. D. Celik, and E. Modiano, "Minimizing transmission energy in sensor networks via trajectory control," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*. IEEE, 2010, pp. 132–141.