

Robotic Mapping and Monitoring of Data Centers

Chris Mansley[†] Jonathan Connell* Canturk Isci* Jonathan Lenchner*
Jeffrey O. Kephart* Suzanne McIntosh* Michael Schappert*

Abstract—We describe an inexpensive autonomous robot capable of navigating previously unseen data centers and monitoring key metrics such as air temperature¹. The robot provides real-time navigation and sensor data to commercial IBM software, thereby enabling real-time generation of the data center layout, a thermal map and other visualizations of energy dynamics. Once it has mapped a data center, the robot can efficiently monitor it for hot spots and other anomalies using intelligent sampling. We demonstrate the robot’s effectiveness via experimental studies from two production data centers.

I. INTRODUCTION

Over time, data centers around the world are consuming ever more energy, with those in the US now responsible for an estimated 2% of the nation’s electricity budget [1], [2]. Today, a single data center rack can be packed with equipment requiring more than 30 kilowatts (kW), ten times what it had been a few years ago [3]. All of the power required to run IT equipment is ultimately dissipated as heat, and a comparable amount of power may be required by the cooling system to remove it. Thus efficient data center cooling is of paramount importance.

A common contributor to data center (DC) cooling inefficiency is over-aggressive cooling, which can result from ignorance of the temperature distribution or from a lack of understanding of how to adjust the cooling such that it is applied evenly, so that all equipment receives just enough cooling, but not too much. The first problem is often addressed by instrumenting the DC with a set of fixed sensors. Previous work [4] has described an effective solution to the second problem, which entails a one-time spatially dense temperature scan of the DC performed by a human pushing around a mobile sensing platform. The dense scan is used in conjunction with a hand-entered map of the DC to provide recommendations on how to better arrange equipment in the center so as to even out the temperature distribution, reducing cooling costs by an average of 10% while ensuring that equipment runs at safe operating temperatures. In subsequent work [5], the same authors address both issues by combining the one-time initial scan with statically placed sensors.

While the one-time scan approach has proven effective in many data centers, a barrier to its broader deployment is

[†]Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA. cmansley@cs.rutgers.edu.

*IBM T.J. Watson Research Center, Yorktown Heights, NY 10532, USA. {jconnell, canturk, lenchner, kephart, skranjac, schapl}@us.ibm.com.

¹An extended abstract, discussing aspects of the data center robot presented here, appeared in the informal proceedings of the 20th Fall Workshop on Computational Geometry, October, 2010, Stony Brook, NY, USA.



Fig. 1. The data center mapping and monitoring robot.

the manual labor and expense of pushing a sensing station around, which can take several days for a large data center (50K sq ft or more). Moreover, given that the workload distribution and the layout may vary over time, it can be advantageous to run the scan frequently.

In order to enable frequent, low-cost DC monitoring, we have developed the fully autonomous robotic platform for navigating, mapping and sensing key environmental data. The robot, shown in Figure 1, has many advantages over previous techniques. It can navigate a DC it has never seen before, simultaneously providing a coarse mapping of the navigable area and providing updates to the temperature map, which are then displayed in real time using a commercial asset management and visualization tool from IBM. The robot provides temperature scans that are more temporally dense than those provided by the mobile sensing station used in earlier work [4], [5] and more spatially dense than can be provided by fixed sensors. Unlike fixed sensors, which are usually mounted on equipment, the robot can sense points in free space. In large data centers that it has already mapped, the robot can selectively choose locations at which to collect sensor readings, so as to maximize information gain while minimizing collection time. Finally, a robotic sensing platform such as ours enables one to easily and cheaply deploy new sensors or sensor types, avoiding expensive installation and configuration costs.

II. RELATED WORK

Several other researchers have discussed the general problem of efficient monitoring in DC environments. Of special

note is the work of Bash et al. [6], who also address selective sampling of sensor locations in a DC. Their technique generates a best set of fixed sensor locations based upon the individual influence of Computer Room Air Conditioning (CRAC) units, as opposed to our technique, which is applied to mobile sensing and hence complements static instrumentation. Also of note is work by Patel et al. [7], [8], who describe automated approaches to monitoring and modeling DC thermal and energy characteristics, and mention robotic monitoring, but do not describe its design or function in detail.

Other researchers have explored robotic approaches to monitoring data in environments other than data centers. Pon et al. [9] describe a cable-based robot for monitoring environmental data in rivers and forests. Stealey et al. [10] extend that work, introducing algorithms for adaptively sampling spatial phenomena [11] that resemble the adaptive sampling strategies used by our robot. Guestrin et al. [12] use the submodularity concept to choose near-optimal sensing paths for multiple robots. While our work shares with these efforts the general goal of efficiently mapping environmental phenomena, their robotic implementations focus on sensing large, unobstructed volumes in rivers or forest canopies. In contrast, our approach incorporates the physical constraints presented by DCs and leverages the domain-specific DC characteristics to provide a targeted end-to-end solution.

III. DESIGN AND IMPLEMENTATION

The primary design objective for the robot is to support frequent, low-cost monitoring of key environmental metrics in a data center. We achieve low cost by

- replacing human labor with a robot;
- constructing the robot from low-cost materials (less than \$700US in total), so that it would be affordable even for modest-sized data centers; and
- endowing the robot with the ability to discover and map the navigable regions of a data center it has never seen before, thereby eliminating the need for humans to provide the robot with a DC layout (which can take several days of labor even in modest-sized data centers).

Another important design criterion for the robot is that it provide sensor data that are most relevant in the data center environment. While we have limited the first deployment to include temperature sensors only, natural next steps include adding air flow and humidity sensors. Since these quantities typically vary vertically, we must provide sensing capabilities at several different heights.

Finally, to help data center administrators grasp immediately the usefulness and practicality of the robot, we designed it to provide realtime feedback to a state-of-the-art DC monitoring and asset management tool: IBM's Maximo for Energy Optimization (MEO). This allows the mapping and monitoring data to be visualized moment-by-moment as it is collected by the robot, and propagated immediately to MEO's built-in analytics, such as dynamic cooling efficiency analysis, cooling zone analysis, and hot spot detection.

A. Platform

The data center robot was developed on top of the iRobot Create robotic research platform. Its robust, low-cost mobile chassis provides a high-level interface for motor control, odometry and a limited number of on-board sensors through a serial connection. The on-board capabilities of the Create platform were augmented with a USB webcam, an off-the-shelf laptop computer and a custom thermocouple interface for sensing temperature. All computation was performed on a netbook with a 1.6 GHz Atom processor and 1 GB of memory, with the exception of the Create's on-board 8-bit processor for motion commands. The thermocouples were attached to a 6 1/2 foot tall extruded aluminum pole mounted vertically to the robot base. The sensors were placed at 1 foot intervals to provide temperature readings at different heights, all the way up to just above the top of the 76-inch racks.

B. Vision

The robot's vision system was designed to exploit the grid-like structure of the tiles in a DC floor. It uses an inexpensive Logitech C905 webcam mounted approximately 30 inches above the floor to image the tile ahead. From the image it extracts the orientation of the robot relative to the floor grid and its location within the current tile. Additionally, the robot uses the vision subsystem to detect different tile structures such as plain and perforated tiles, inherent to DC design. There are certain challenges associated with the vision process due to extraneous lines introduced by boxes and cabinets, different patterns exhibited by perforated tiles and the occlusion of some tile borders by DC equipment. Our approach, as presented here, addresses these challenges to provide a robust vision-based DC mapping, positioning and navigation technique for autonomous robot operation.

The basic mode of operation is for the robot to traverse the DC, stopping at each tile to take temperature readings. Yet, due to imprecise mechanicals and surface irregularities, sole reliance on odometry is not adequate [13]. After stopping in a tile the robot grabs a 320x240 color image and processes it to determine its actual pose, whether the next tile is *visitable* or *blocked*, and its tile type (i.e., *perforated* or *plain*). The image processing takes place in four steps: (a) grid line finding, (b) robot pose determination, (c) visitability check, and (d) tile type detection. These steps are highlighted in Fig. 2, and further described below.

Grid Line Finding: This first step pre-emphasizes borders (and suppresses perforations) using a rectangular min filter. It then finds Sobel edges above a threshold (Fig. 2(a)) and groups them into 4 major directions. A connected component analysis is done to remove small edges as well as insufficiently elongated regions. Edge groups are then approximated by straight lines and re-projected onto a flat floor using the calibrated camera geometry.

Robot Pose Determination: The angles of the detected lines are histogrammed (modulo 90°) and the peak of the histogram is used to determine the overall grid and robot orientation. The system then removes all lines that are not close

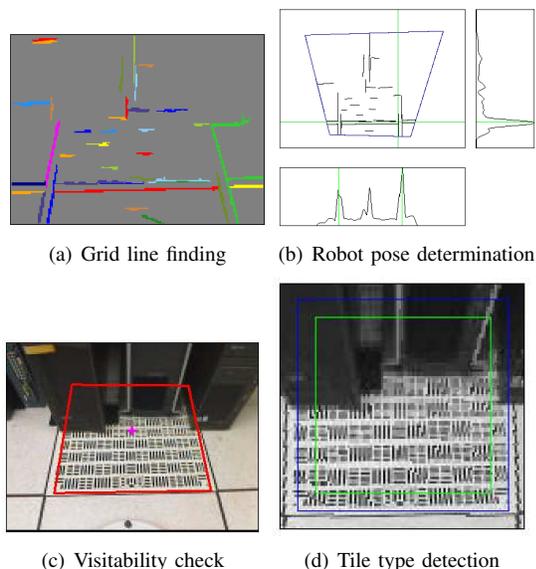


Fig. 2. The vision process overview.

to the inferred grid orientation and projects the remainder along the x and y axes. Using one or two major peaks in each projection, the system determines the boundaries of the tile ahead and the robot’s relative offset from the tile center. For example, the two histogram peaks in Fig. 2(b) clearly identify the lower left corner and the boundaries of a tile.

Visitability Check: At this point, the image boundaries of the next tile have been established, so the system remaps the corresponding portion of the input image to a medium-sized (104×104) floor-based representation. The Sobel edge finder is again applied and the periphery is checked for largely continuous edge energy. Lack of a clearly bounded edge on some side is taken as evidence of overlapping obstruction. Fig. 2(c) shows such an example, where the projected tile boundary is obstructed by a rack, and thus the tile is marked as blocked. The system also checks the center portion of the tile for the numbers of edge pixels and changes in intensity to determine whether a tile is visitable or not.

Tile Type Detection: In the case of perforated tiles, the edges in the center of a tile are simply due to perforations used for venting cool air into the data center, and the robot can safely cross such a tile. To detect perforated tiles (Fig. 2(d)), the system first attempts to normalize for inter-tile intensity variations by comparing each pixel to the average in a small (17×17) surrounding region. It then computes the average and standard deviation of this “dark energy” to make sure the potential perforations are roughly symmetric and evenly distributed. As a final test the system re-centers and further subsamples the tile image to create a small (11×11) “signature image”. This signature image is rotated by 90° , 180° , and 270° and compared with the original. The maximum pixel difference with any version is then checked to assure that the pattern is symmetric, indicating perforation rather than an obstruction.

The robot’s vision system has certain advantages relative to more sophisticated techniques such as visual SLAM (Simultaneous Location and Mapping). First, since the robot

only pays attention to the floor grid, it is insensitive to re-arrangements of the racks and furniture. Second, it directly observes tiles and classifies them accurately as traversable or blocked, and perforated or plain. With visual SLAM, there is no guarantee that obstructions would always generate salient features that enabled such classification.

C. Planning, Localization and Mapping

The resulting vision system is very robust with respect to contrast, illumination and partial occlusion of the grid. However, it only provides a differential pose relative to a tile, rather than a full robot pose with respect to the room. Odometry must be used to keep track of tile crossings and major rotations (e.g. 90° degree re-orientations). For robot global localization, we combined these differential visual observations with odometric feedback from the Create platform using an Extended Kalman Filter (EKF) [14]. The global reference frame is defined by the starting location and orientation. The rotation and translation of the frame are handled later in the analytics software. We assume that the DC forms a regular grid. While these simplifying assumptions do not solve the “kidnapped robot problem” [14], the lower-cost and simplicity of the system outweigh the added benefits of a complete SLAM solution.

Given the current tile pose in the global reference frame, the map containing the tile type (perforated or not) and tile obstruction (occluded or not) must be created. In order to fully explore the data center layout, we use a frontier-based exploration system [15]. The exploration was performed by navigating the robot to the nearest unvisited location using the A* search heuristic [16]. In the simulations of section V, this exploration algorithm visited only 10% more tiles than the number of visitable tiles in the DC.

D. Thermal Monitoring

The temperature sensors are a 40-gauge K-type thermocouple that has approximately a 3-second response to temperature changes. The sensors are placed starting just a few inches above the ground and then at six one foot increments to give a reasonably fine-grained measurement of the temperature stratification at different heights. The electronics circuit board contains circuitry that converts the thermocouple sensor voltage into a scaled voltage, which is sampled by an analog-digital converter built into a microcontroller. The microcontroller samples each sensor in sequence and transmits the data for subsequent storage and analysis.

E. Product Integration

As the robot incrementally maps and monitors the data center, it wirelessly sends realtime updates to MEO, which graphically displays a continually growing layout and thermal map. MEO uses a different graphic to represent each tile type as it is classified by the robot: either standard (non-perforated), perforated, blocked or unknown. Since the robot does not distinguish the nature of the objects that are resident on occupied (and therefore obstructed) tiles, the

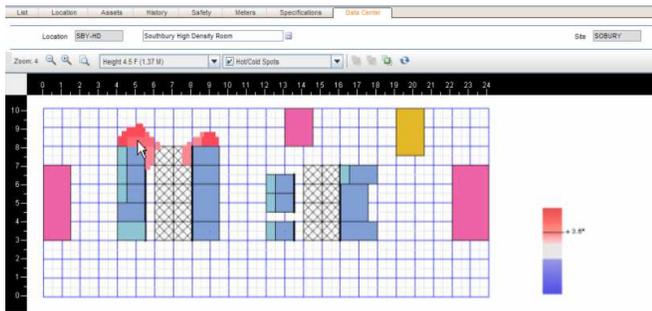


Fig. 3. A hot/cold spot map from the Southbury data center in MEO. Hot air is spilling around the sides of racks at the end of the left hand cold aisle creating two hot spots. The red hot spots indicate that temperatures exceeds the high temperature threshold in the vicinity of the air inlets of equipment.

result is a somewhat coarse but still useful and recognizable representation of the data center.

If MEO has already been populated with the DC asset information, then it can display this layout and simply update the temperature map as it is discovered incrementally by the robot. In this case, the data collected by the robot can be fed to MEO’s built-in analytics, and the results can be rendered graphically by MEO. One example is MEO’s hot spot analysis and visualization capability. Since certain regions of the data center are more sensitive to high temperatures than others (for example the air inlets of IT equipment, as opposed to the exhaust), MEO computes differential high-temperature thresholds across the data center and highlights regions that exceed the local threshold, as illustrated in Figure 3. MEO’s hot and cold spot detection is a key starting point for effective adaptive cooling [17].

IV. FIELD DEPLOYMENTS

We conducted two sets of experimental deployments of the robot in two separate production DCs. The first set of experiments was performed at a research DC in Hawthorne, N.Y. This research DC presented many interesting challenges not typically found at IBM’s non-research DCs due to highly-frequent reconfiguration and provisioning operations. Perhaps most importantly, this DC used many different types of perforated tiles, making it an ideal site for training the robot’s tile classifier. We conducted dozens of experimental runs in this DC to tune the vision and localization algorithms and to test the MEO integration.

We mention two of the more interesting lessons learned from these experimental runs: In the early days of the vision tuning we thought that it would be sufficient to look just for one tile boundary and a clear tile ahead. However, with just single edge detection, the robot was confusing certain white power distribution units (PDUs) with clear non-perforated tiles. Later on, the robot started to mistake a variety of different perforated tiles with few exemplars for obstacles and we realized that symmetry checking need to be an important part of tile type detection.

Our experiments demonstrated the robustness of the robot in detecting and navigating through a variety of obstacles that can be encountered in DCs, the reliability of localization

with dynamically-varying DC layouts, and the resilience of our vision system in classifying a range of different tile configurations. Overall, this Hawthorne DC comprised 4800 square feet, contained 220 visitable tiles and was completely scanned in 46 minutes.

The second of our deployments was at an enterprise DC in Southbury, CT. This DC was much more static, tiles were relatively uniform, and obstructions were limited in number. Although the tiles in this DC differed substantially from those of the research DC, with different colors, perforation patterns and tile edges, the robot successfully scanned this data center on its first run and in several follow-up runs. This DC comprised 960 square feet, contained 115 visitable tiles and our initial scan was completed in 28 minutes.

V. SELECTIVE SAMPLING: ALGORITHM, SIMULATION AND PHYSICAL EXPERIMENTS

In small DCs such as those used in field deployments, complete scans of every location will complete in a reasonable amount of time. However, in large DCs, complete dense scans could take days to complete, rendering continuous monitoring impractical. A dense scan over a long time period could have inherent temporal variations, reducing the scan’s usability. Moreover, in many enterprise DCs, administrative constraints would limit the duration for actively scanning the DC. Thus, in lieu of repeat dense scans, it is desirable to judiciously select a number of informative points that accurately represent the thermal profile of the entire DC.

A. Selective Sampling with Gaussian Processes

The goal of selective sampling is to accurately model the DC temperature profile with a reduced set of most informative sensing locations. To derive the overall profile from selected samples, we used an approach based on non-parametric models to represent the DC profile as Gaussian Processes [18]. Using this class of probabilistic models, the uncertainty of our predictions can be quantified using the measurements already taken. These uncertainty estimates allow us to select additional sensing locations to minimize the error in the predicted profile. Computing the optimal locations is non-trivial and is dependent on the metric used to quantify the uncertainty of adding sensing locations. One naive approach minimizes the uncertainty (variance) of the measurement locations by greedily choosing the locations that reduce the variance most. Greedy variance minimization has been shown to only indirectly improve the error of the fitted function. In our work, we used the mutual information (MI) criterion proposed by Guestrin et al. [11] to determine a near-optimal subset of sensing locations, which we further describe below.

Our environment has discrete sensing locations, one per floor tile. The set of sensed locations, \mathcal{A} , is the set of locations already measured by the robot. Our objective is to maximize the change in mutual information by adding a new sensing location. More formally, we wish to maximize:

$$MI(\mathcal{A} \cup y) - MI(\mathcal{A}) = H(y|\mathcal{A}) - H(y|\bar{\mathcal{A}}) \quad (1)$$

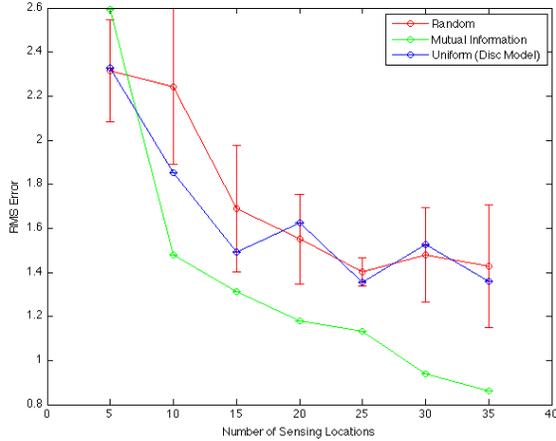


Fig. 4. Root Mean Square (RMS) errors for different sparse sampling strategies and different numbers of sensing locations

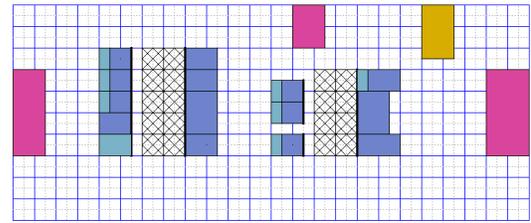
where $H(x|z)$ is the conditional entropy and y is the proposed sensing location. This is done iteratively by adding a sensing location to the set \mathcal{A} , which maximizes Equation 1. Computing this conditional entropy is simplified by the choice of Gaussian Processes. Notice that the computation includes the conditional entropy at unobserved locations as well as observed locations. This MI criterion measures the reduction in uncertainty at unobserved locations in contrast with variance minimization, which only looks at the conditional entropy given the sensed locations.

Specifically, we learn a squared exponential covariance function with automatic relevance determination (ARD) from the data in the dense scan using marginal likelihood maximization [18]. This covariance function provides a different hyper-parameter for each dimension allowing x and y dimensions to have distinct weights in the distance function.

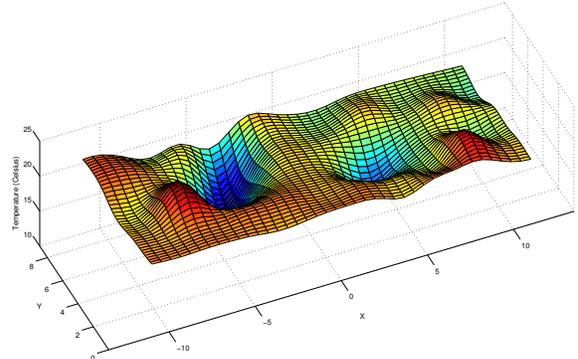
B. Simulations

We simulated the sensing and movement of the robotic system to evaluate the speed and accuracy of reconstructing the thermal profile using dense and sparse scans. The thermal profile was constructed as an arbitrary function with Gaussian hotspots and Gaussian noise at each sampled location. An actual medium-size DC in Zurich, Switzerland containing approximately 500 tiles was used for the evaluations. For each experiment, a complete dense scan was performed, the hyper-parameters of the Gaussian Process regression were learned, and then k additional measurements were allowed. These k additional measurements were used, in conjunction with the previously learned hyper-parameters, to create an approximation of the thermal profile. The regression and the original function were compared on a regular grid of points and the root mean squared error is reported. Random algorithms were run 5 times for each value of k , with the mean and standard deviation reported.

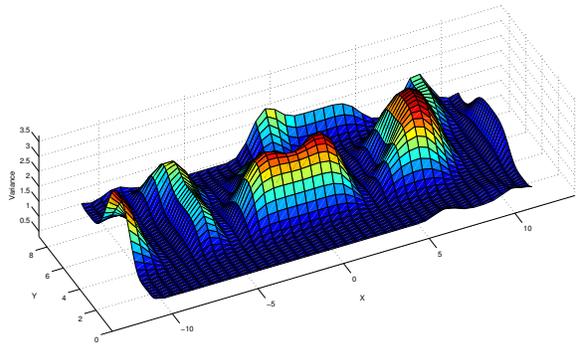
Results for three algorithms are shown in Fig. 4. The first randomly selects sensing locations. Given k additional sensing opportunities, the algorithm selects k points at random from all possible sensing locations. The second algorithm is



(a) Data Center Layout



(b) Temperature



(c) Uncertainty

Fig. 5. The layout for the production data center with the measured thermal profile function approximation and the uncertainty of this estimate

an approximate uniform coverage algorithm. It picks k “well separated” points from amongst the dense set of candidates, where the radius used to determine what constitutes a well separated set is obtained from disk packing considerations. The third algorithm selects sensing points according to the mutual information criterion described previously. The only information used from the dense scan in the sparse scan is the hyper-parameters defining the Gaussian Process and the possible set of sensing locations. As seen in the figure, picking just 15 sensing locations using MI gives an RMS error that is consistently less than the error for both uniform and random sampling for any number of samples.

C. Physical Experiments

We also validated our selective sampling strategy with actual experiments in the Southbury DC. Here we first configured the robot to create a complete dense scan of the entire DC. The temperature profile acquired from this initial scan—interpolated using Gaussian Process regression—and the associated uncertainty at different DC locations are shown in Fig. 5(b) and 5(c) respectively. Notice the areas of high

uncertainty correspond to the locations of the server racks and equipment in the DC (Fig. 5(a)), where no measurements could be taken.

After the initial scan, we configured the robot to operate in “selective sampling mode”, where it computed 10 selective sensing locations, and traversed a path along these 10 points to quickly collect temperature samples at these specific locations. This path is computed by greedily navigating to the closest sampling location using A* search, measuring the temperature, and repeating the process. Then, we again used Gaussian Process regression to model the entire DC thermal profile by only using the sensor readings from the selected 10 points.

Similar to the simulations, we used the dense scan information as the ground truth and compared the selective sampling interpolation with this to evaluate our accuracy. We repeated the same experiment twice during different operating periods of the DC. The RMS errors achieved with each of the two sampling evaluations are 1.72 and 2.27°C, respectively, where part of this error is also attributed to the inherent temporal thermal variations present in the physical environment. These results show, in an actual enterprise DC setting, that the robot can effectively model entire DC thermal profiles with reasonable accuracy using selective sampling. It is worth noting that in this evaluation it took the robot only 5 minutes of monitoring to collect the sparse sensing information and to map the entire DC thermal profile.

VI. CONCLUSION AND FUTURE WORK

We have described a fully autonomous, low cost, and robust mobile robotic platform for navigating and mapping an unknown data center while simultaneously monitoring the environment. We have integrated the robot into Maximo for Energy Optimization, an enterprise-level asset and data center management product from IBM that provides visualizations of energy dynamics and other analytical tools that support effective energy management. In this capacity, the robot (1) provides spatially-dense data center scans at lower cost than human-aided methods, enabling them to be run much more frequently, and (2) responds quickly to dynamically-emerging cooling problems that could lead to equipment failure or wasteful use of energy.

In designing the robot, we exploited data center domain characteristics such as tile-based floor structures to enable a low-cost and robust vision-based navigation and tile detection system. Our data center experiments showed that frontier-based exploration can complete a full scan with only 10% more tile visits than there are visitable tiles. We have validated the robot’s operation in two production data centers, one of which had never been seen before. In both cases, the robot successfully scanned and provided a complete thermal and layout map in less than an hour. To address potential time constraints in larger data center scans, we introduced a selective sampling approach based on mutual information. Our evaluations with both full-scale simulations and field experiments on multiple data center layouts showed that such intelligent selective sampling can

keep scanning times to a small fraction of the time for a full scan and achieve DC temperature profiles with errors on the order of 2°C or less.

In the near future, we will be extending the platform to include air flow and humidity sensors. We will also be taking the robot to some of the largest IBM data centers where we will do larger scale testing of our MI-based selective sampling. We will evaluate the effectiveness of the naive incremental path planning algorithm we currently use for selective sampling and explore alternatives, such as those introduced by Guestrin et al. [12], or more traditional traveling salesman problem approximations.

REFERENCES

- [1] J. G. Koomey, “Estimating total power consumption by servers in the U.S. and the world,” 2007.
- [2] U.S. Environmental Protection Agency ENERGY STAR Program, “Report to congress on server and data center energy efficiency,” 2007.
- [3] Gartner Inc., “Gartner Says 50 Percent of Data Centers Will Have Insufficient Power and Cooling Capacity by 2008,” Press Release, November 29, 2006.
- [4] H. F. Hamann, M. Schappert, M. Iyengar, T. van Kessel, and A. Claassen, “Methods and techniques for measuring and improving data center best practices,” in *Proc. of 11th Intersoc. Conf. on Thermomechanical Phenomena in Electronic Systems*, May 2008, pp. 1146–1152.
- [5] H. Hamann, T. van Kessel, M. Iyengar, J.-Y. Chung, W. Hirt, M. A. Schappert, A. Claassen, J. M. Cook, W. Min, Y. Amemiya, V. Lopez, J. A. Lacey, and M. O’Boyle, “Uncovering energy efficiency opportunities in data centers,” *IBM J. of Rsch & Dev*, vol. 53, no. 3, pp. 10:1–10:12, 2009.
- [6] C. E. Bash, C. D. Patel, and R. K. Sharma, “Dynamic thermal management of air cooled data centers,” in *Proc. of the 10th Int’l Conf. on Thermal and Thermomechanical Phenomena in Electronics Systems (ITHERM)*, San Diego, CA, May 2006, pp. 445–452.
- [7] C. Patel, C. Bash, R. Sharma, A. Beitelmal, and R. Friedrich, “Smart cooling of datacenters,” *Proc. of the PacificRim/ASME Int’l Electronics Packaging Tech. Conference and Exhibition (IPACK)*, July 2003.
- [8] C. Patel, “A vision of energy aware computing from chips to data centers,” *Proc. of the International Symposium on Micro-Mechanical Engineering (ISMME)*, Dec 2003.
- [9] R. Pon, M. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W. Kaiser, M. Srivastava, G. Sukhatme, and D. Estrin, “Networked infomechanical systems: a mobile embedded networked sensor platform,” 2005.
- [10] M. Stealey, A. Singh, M. Batalin, B. Jordan, and W. Kaiser, “Nimsaq: A novel system for autonomous sensing of aquatic environments,” 2008.
- [11] C. Guestrin, A. Krause, and A. P. Singh, “Near-optimal sensor placements in gaussian processes,” in *Proceedings of the 22nd International Conference on Machine Learning*, 2005.
- [12] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin, “Efficient planning of informative paths for multiple robots,” in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007.
- [13] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, “Mobile robot positioning: Sensors and techniques,” 1997.
- [14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
- [15] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *In Proceedings of the IEEE International Symposium on Computational Intelligence, Robotics and Automation*, 1997.
- [16] P. Hart, N. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [17] R. Das, H. Hamann, J. Kephart, and J. Lenchner, “Utility-function-driven energy-efficient cooling in data centers,” *Proceedings of the 7th International Conference on Autonomic Computing (ICAC 2010)*, pp. 61–70, 2010.
- [18] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.