An Application of Laser-Based Autonomous Navigation for Data-Center Monitoring

Stefano Rosa[1], Ludovico Orlando Russo[1], Giuseppe Airó Farulla[1], Luca Carlone[2], Roberto Antonini[3], Gaspardone Marco[3], Basilio Bona[1]

[1] Politecnico di Torino, Dipartimento di Automatica e Informatica
{<name>.<surname>}@polito.it
[2] Georgia Institute of Technology, College of Computing
luca.carlone@gatech.com
[3] Telecom Italia
{roberto1.antonini,marco.gaspardone}@telecomitalia.it

Abstract. Data-center monitoring has been a critical subject of research in recent years. Mobile robots have been successfully employed in the industrial field to efficiently perform common tasks. In this paper we report some preliminary results on the study and development of a robotic system, in which a mobile robot equipped with a laser range sensor and an IMU is able to autonomously navigate in a data-center room for accurate monitoring of critical measurements, such as servers' external temperature, humidity and other physical quantities. The robot is able to autonomously create a map of a previously unknown room, localize therein and execute a list of measurements at different locations, which are provided by the user via a web graphical user interface (GUI). The robot is able to find the best trajectory to reach the given locations, while avoiding static and moving obstacles. The particular characteristics of the data-center scenario introduce specific problems related to map creation and localization using laser-based techniques (e.g., irregular surfaces as metal grids and high symmetry of the environment), which must be properly taken into account and are discussed throughout the paper. Preliminary experimental results show that the system is able to create a consistent map of the environment, to correctly localize itself therein and to follow a given path.

Keywords: mobile robotics, data-center monitoring, SLAM, localization, path planning

1 Introduction

Service robotics refers to all robotics applications aimed at assisting or substituting human activities in tasks that are dangerous or repetitive and that do not require human reasoning. Autonomous navigation techniques for mobile service robotics have been widely studied in the last two decades [1].

Although laser-based state-of-the-art solutions for the different navigation problems are well known in literature, few applications of these techniques have been proposed in consumer or industrial environments, such as the case proposed in [2].

In this paper, we propose an application of these techniques to the well known problem of data-center monitoring. The data-center environment is a good application scenario for a mobile robotic service: it is a mostly static environment, presenting few moving objects and people. Moreover, data-center monitoring is a problem that has not yet been completely solved. In fact, the scenario presents some challenges, such as its symmetry and irregular surfaces. In particular, server racks are usually covered with metallic grids, which are not well detected by a laser range finder. On the other hand, vision-based systems have not yet proved to be reliable enough for long-term industrial applications.

Data-center administrators have to guarantee privacy, robustness and availability of all servers at any time [3]. Due to the dimensions of a data-center room, monitoring and diagnostics pose a very big challenge. In fact, data-centers are among the most intensive energy consumption buildings [4, 5, 6]. In particular, energy consumption goes to cooling systems that maintain the machines in safe working conditions. In order to save energy, temperature is often kept as high as possible. This increases the risk of hot spots and thus requires better accurate monitoring of temperature and humidity. The dimensions of a typical data-center room, makes it practically impossible instrument the whole environment. As of now, a solution is to monitor the perimeters of the room, while other parts are periodically inspected by human operators. This increases management costs.

This paper is organized as follows: in Section 2 we present some existing solutions for data-center monitoring; in Section 3 we describe the proposed solution in detail; in Section 4 we present some preliminary results about navigation performances in a real data-center scenario. Finally, in Section 5 we draw some conclusions and present future activities.

2 Related Works on Data-Center Monitoring

Data-center monitoring refers to all the solutions aimed at measuring environment variables (usually temperature and humidity) in data-center rooms for monitoring and diagnostics. It has been the subject of several research activities in recent years, devoted to reduce cooling and management costs, but at the same time to reduce machine malfunctions caused by hot spots. Traditionally, data-center rooms are controlled using Computer Room Air Conditioning (CRAC) units aimed at pressurizing a room with cool air. In each room, servers are organized in racks. A single server room contains hundreds of racks, each containing tens of servers.

Early works about data-center energy management have been proposed in [7, 8]. Some of the authors identify inefficiencies in data-center energy management and propose a smart cooling system based on localized cooling and a distributed metrology layer to sense and control temperature and pressure. Other works about efficient monitoring and control are proposed in [9, 10, 11, 12, 13]. However, most of these approaches require to instrument the data-center with a complex sensing network and integrated control knobs. These solutions come with two big problems, that are scalability and management. In particular, the bigger the size of a data-center is, the more expensive is the sensing and control interface. Moreover, periodical maintenance, verification and calibration are required to ensure the perfect working. Even if some of the cited solutions propose sensing topology that minimized the number of required sensors to monitor the whole room, it is clear that scalability cannot be achieved using a static monitoring network.

Mobile solutions have been proposed in order to solve scalability and management problems. For instance, [14, 15] present a Mobile Measurement Technology (MMT) with a mobile sensor cart, which can optionally be integrated with static sensing, for spatially-dense data-center monitoring.

Recently, the use of robots has been proposed. In [16] the authors propose a multi-robot system for data-center monitoring, where robots communicate each other wirelessly. Localization is based on NFC Tags placed on the floor. However, this solution requires physical intervention on the environment, i.e., placing the NFC tags. The system does not adapt to changes in the racks position neither is able to work in not instrumented rooms. Moreover, this solution does not provide obstacle avoidance to allow the robots to safely move when people are operating in the room.

In [17] the authors propose to use a low-cost autonomous robot for data-center monitoring. In this solution the robot takes advantage of the standard square tiles on the data-center floor for navigation. The robot tries to visit all unobstructed tiles of the data-center. When it visits a tile, it stops in its center and takes measurements. The solution guarantees a complete coverage of the data-center room. The solution is not able react to moving obstacles, neither to plan an optimized trajectory in order to measure only a subset of interesting locations. Moreover, the full coverage approach is too slow to guarantee a good monitoring of each area. Finally, this approach is strongly sensitive to the tiles' color: it is not suitable with dark tiles as it would be very difficult to recognize edges.

Differently from other mobile and robotic approaches, our solution guarantees is a complete reconfigurable system, that does not require any ad-hoc intervention on the data-center room. Moreover, the use of state-of-the-art navigation solutions allows the robotic platform to work also in dynamic environments when human operators are freely moving in the data-center room, and also when changes in the topology of the room occur.

3 The Proposed Solution

This Section describes the proposed solution. The Section is so organized: Section 3.1 summarizes the characteristics of the application scenario and introduces some related issues; Section 3.2 analyzes the main architecture of the system; Section 3.3 discusses the map creation phase, which is necessary to set up the system; finally Section 3.4 discusses the localization algorithm.

3.1 Application Scenario and Related Issues

The main goal of the proposed solution is the development of a relatively low-cost robotic system characterized by high flexibility, i.e., which is able to operate in an environment that can be initially unknown without any setup procedure or ad-hoc intervention. For this reason, it is essential to analyze the main characteristics of the working scenario.

The particular characteristics of data-center spaces introduce specific problems that must be tackled in order to assure safe and reliable laser-based autonomous navigation. Racks containing servers are covered by a metal grid that introduces noise into laser scanner distance measurements. The noise is due to the fact that laser beams sometimes go through the holes of the grid. This phenomenon introduces errors in mapping and localization. Moreover, new racks are constantly added or removed, so that service robotic applications in data-center monitoring areas have to deal with intrinsically dynamic environments.

Since the environment is initially unknown, the robot must be able to autonomously create a map of the environment to be used for localization. High symmetry of data-center rooms (data-center areas are organized as regular grids, with rows of racks and corridors) poses issues for localization, and any localization algorithm must be able to handle different localization hypotheses over time.

Finally, since the rooms in which the system operates can be very large, the algorithm should be suited for large-scale environments.

3.2 System Architecture

This Section illustrates the main architecture of the proposed solution. The main functional blocks are shown in Figure 1 and described hereafter.

The proposed solution is based on two memory blocks: the **Map Database** and the **WP Database**. The **Map Database** contains all the available maps obtained by performing SLAM within each data-center room. When a map is available, the user is able to set *waypoints* within the map, i.e., interesting points where the robot have to perform some *task*, defined by the user too. Tasks are actions that the robot have to perform at a specific waypoint, such as recording temperature or capturing a thermal image of the nearby racks. To each waypoints, a set of tasks to be performed is associated. The user is also allowed to set forbidden areas within the map, where the robot is not allowed to go into. Forbidden areas, waypoints with associated tasks and the corresponding data which is recorded by the robot are stored in the **WP Database**.

The **Web User Interface** is a web-based graphical user interface (GUI) which shows the current map and the data stored in the **WP Database** to the user, together with the actual position of the robot within the map, obtained by the **Localization** block. Using the web interface the user is also able to create new paths, by adding and deleting waypoints and associated tasks. An example of the GUI is shown in Figure 2.

The **Task Manager** block is in charge of managing the path that the robot has to follow. The path is composed by a list of waypoints allocated in the

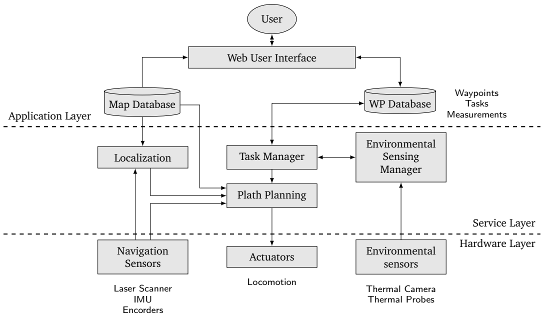
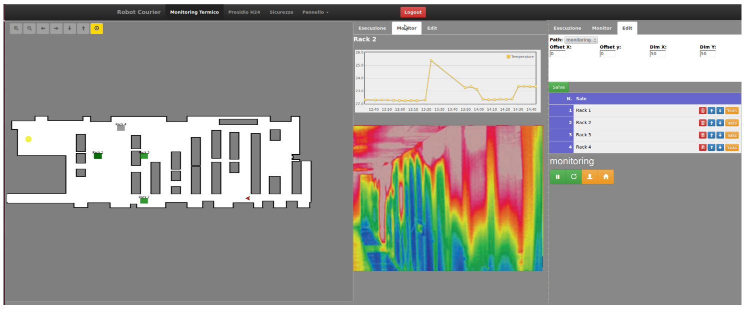


Fig. 1. The architecture of the proposed solution.



Rack 2

Path: monitoring

Offset X: 0

Offset y: 0

Dim X: 50

Dim Y: 50

N.	Sale
1	Rack 1
2	Rack 2
3	Rack 3
4	Rack 4

monitoring

Fig. 2. The Web User Interface. On the left, the data-center room with some user-defined waypoints. In the middle, temperature data and thermal image for the current waypoint. On the right, the list of waypoints and associated tasks.

WP Database. Different tasks can be executed by the robot at each waypoint. Finally, the block is in charge of collecting environmental data, captured by the robot, and store them in the **WP Database**.

The **Environmental Sensing Manager** is in charge of collecting measurements from the environmental sensors (e.g., *thermal camera* or *thermal probes*) when they are required, and to transmit them to the **Task Manager**.

The **Localization** block performs localization and position tracking of the robot within the map. It uses information coming from the **Navigations Sensors**, i.e., the *Laser Scanner*, the *Inertial Measurement Unit* (IMU) and the wheel *Encoders*.

Finally, the **Path Planning** block is in charge of finding the best trajectory to reach each new waypoint. It receives as input the map of the environment, sensors data, the estimated position of the robot and the next goal. It generates a global trajectory and a local trajectory, which outputs velocity commands to the rover. The global trajectory is the minimum distance path to reach the next goal by avoiding static obstacles (racks) and forbidden areas; the local trajectory is generated on the fly as the robot is moving by detecting moving obstacles with the *laser scanner* by creating a rolling window around the position of the robot.

The whole system has been developed using the Robot Operating System (ROS) [18]. The different blocks are implemented as ROS nodes and can reside onboard of the robot as well as remotely.

3.3 Map creation

Since the scenario in which the robot has to operate can be initially not known, the ability to autonomously create a map of the environment is required. The problem of map estimation in an unknown environment has been treated extensively in robotics, and the corresponding framework is usually referred to as *Simultaneous Localization and Mapping* (SLAM). Mapping approaches are required to be accurate, since a faithful representation of the environment is necessary for robot localization and planning, and scalable, since a robot should be able to map also medium and large-scale environments. In order to cope with these requirements, in our system we use a novel solution to the SLAM problem which is called *pose graph optimization* [19, 20, 21]. In pose graph optimization, each node represents a pose assumed by a mobile robot at a certain time, whereas an edge exists between two nodes if a relative measurement (inter-nodal constraint) is available between the corresponding poses. Inter-nodal constraints are usually obtained by means of proprioceptive sensors (*odometry*) or exteroceptive sensor-based techniques (*scan matching*, etc.). Given these contraints, the objective of pose graph optimization is to estimate the configuration of the nodes' poses (*pose graph configuration*) that maximizes the likelihood of inter-nodal measurements.

Odometric edges are built starting from the results of a combination of odometry and scan matching (the translation estimation is given by odometry, while the angular rotation estimation is given by scan matching), since in a corridor-like environment the translation estimated by odometry could be more accurate than the one estimated by scan matching. At each step the algorithm also checks for possible loop closings using the same scan matching algorithm between the current pose of the robot and all the candidates taken from the past poses. For scan matching we use the *Canonical Scan Matcher* (CSM) algorithm described in [22]. Each time one loop closing is found, we optimize the pose graph using the linear pose-graph optimizer presented in [23] called LAGO (Linear Approximation for Graph Optimization). A number of consistency checks is implemented during the constraint creation as well as after the optimization in order to discard wrong constraints.

3.4 Localization

We use a modified version of the *Adaptive MonteCarlo Localization* (AMCL) algorithm, originally proposed in [24], implemented in ROS. The algorithm maintains a probability distribution over the set of all possible robot poses, and updates this distribution using data from a laser scanner. The algorithm requires a pre-defined map of the environment against which to compare observed sensor values.

At the implementation level, the AMCL code represents the probability distribution using a particle filter. The filter is "adaptive" because it dynamically adjusts the number of particles in the filter: when the robot's pose is highly uncertain, the number of particles is increased; when the robot's pose is well determined, the number of particles is decreased. This is a trade-off between processing speed and localization accuracy.

Global localization In our version of the algorithm, global localization is also possible, versus the original ROS implementation, which only implemented position tracking. Global localization represents the problem of retrieving the robot pose starting with no information on its initial pose. The particles of the particle filter are thrown all over the empty regions of the map with uniform probability. After an initial settling phase the algorithm then switches to position tracking.

Sensor fusion (IMU-corrected odometry) Another key improvement we made is the fusion of wheel odometry and inertial measurement units for improving the odometric trajectory estimation of the robot.

The problem of odometry drift is well known in the literature. Wheel odometry, in particular when using 4 wheels drive rovers, is affected by large errors, due to wheel slipping, encoders precision and approximations in the velocity estimation. In our experiments the Corobot 4WD robot showed large odometry drift, in particular when the robot is turning. This is due to both approximations in the internal odometry estimation and wheel slip. Experiments showed that, due to this drift, AMCL localization was prone to failures, in particular when the robot was turning.

We used the angular velocity measures given by a commercial IMU and included them into the estimation of the velocities of the robot, by integrating them alongside the measures given by wheel odometry. Since gyros are affected by bias errors, we also do bias estimation by averaging their value whenever the robot is not moving. We experimented with the internal IMU of the robot, but we found out that its accuracy in heading estimation was not sufficient. So we added an external IMU. This sensor, in combination with the developed ROS node, performs better and is able to provide corrected heading estimation to the localization algorithm.

3.5 Path planning

For autonomous mapping we developed a ROS implementation of the *Virtual Field Histogram* (VFH+) obstacle avoidance algorithm [25]. This particular implementation is tailored for use with a laser scanner. In our implementation the robot always goes forward, while avoiding static and moving obstacles. This behavior is usually called "wandering".

For the actual path planning during normal operation we used the *move_base* package provided by ROS. It is a flexible two-level approach which is composed by a global path planner based on Dijkstra graph search that is in charge of generating a feasible trajectory from the robot position to the next goal, and

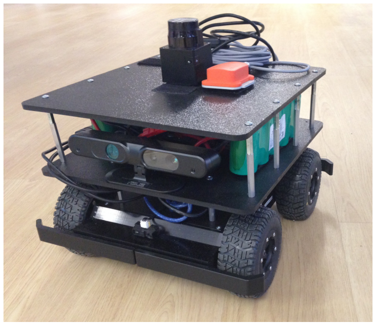


Fig. 3. The robotic platform used in our application.

a local path planner which implements obstacle avoidance using the *Dynamic Window Approach* (DWA).

4 Early Stage of Validation

This Sections presents preliminary results aimed at validating the proposed solution when working in a real environment.

In our experiments, the Corobot Classic platform from Coroware was used. It is a 4WD rover endowed with a variety of sensors, in particular: wheel encoders, an internal IMU, a webcam, bumpers. A Hokuyo URG-04LX laser range finder was added. We also used an external XSens MTI inertial measurement unit, which proved to be more accurate than the internal one. The sensors used for monitoring are two thermal probes at different heights and a thermal camera. The robot is provided with homing capabilities to a docking station. In Figure 3 an image of the used robot platform is provided.

The scenario for our application is a data-center room, composed by free corridors and several rows of racks. The environment for this particular experiment is challenging as most of the server racks are covered by grids, introducing large errors in range measures, as laser rays are passing through the grids introducing measurement errors. Moreover, the symmetry of the environment poses a challenge for loop closings detection.

We focused our preliminary experiments on testing the autonomous navigation performances. This task is of relevance because it ensures on the feasibility of the entire system. Hence, the rest of this Section reports results demonstrating the ability of the robot of correctly mapping the environment and localizing within also when running during long time periods.

4.1 Mapping results

We tested our algorithm using two different laser range sensors: an Hokuyo URG-04LX and a SICK LMS-200. The Hokuyo has an angular range of 240° with an

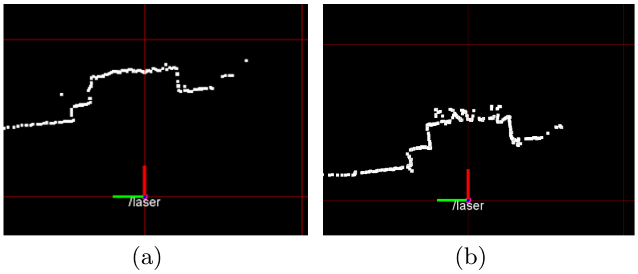


Fig. 4. Effect of the grids on the laser range readings. (a) SICK LMS-200 (b) Hokuyo URG-04LX. Due to better angular resolution, the Hokuyo is affected by measurement noise due to metal grids covering the racks.

angular resolution of 0.46°, but has a maximum distance range of $4.5m$. The SICK LMS-200 is more expensive, covers only 180° of range, with a resolution of 0.5° or 1° and has a maximum usable range of $10m$, with an absolute maximum range of $32m$ and an error of $10mm$ at $32m$. Before the final tests, feasibility tests have been carried out to test the performances of laser range finders in presence of challenging obstacles as the grids that can be encountered in the real data-center. We found out that only the SICK laser scanner was able to give reliable distance measures, while the Hokuyo laser scanner showed noisy data (Figure 4).

We then tested our SLAM approach on a typical data-center room. In our approach the robot did not perform active exploration; instead it was moving according to a simple wandering behaviour. Most of the server racks were covered by metal grids, thus making the scan-matching part prone to errors, as the laser rays were sometimes passing through the grids introducing big measurement errors. Moreover the symmetry of the environment posed a challenge for loop closing detection. Finally, the presence of slippery grids on the floor introduced large odometric errors. Nevertheless, the algorithm was still able to create a consistent map of the data-center room. The result is shown in Figure 5.

4.2 Localization and path planning results

We tested localization and path planning inside the same room, using the map created by SLAM. The robot was traveling during normal daily operations, in presence of workers and changes in the environment (open rack doors, equipment). In each experiment a path composed by a large number of waypoints was created. The tasks associated with each waypoint were thermal camera image acquisition and subsequent temperature measurement. Figure 6 shows the results of a typical experiment. It can be noted that the robot correctly localized itself and was able to follow the given path. Ground truth was not avaiable for estimating localization accuracy, but Figure 6 shows how the covariance of the pose estimate from the particle filter remains small over the whole experiment. We also measured localization accuracy. The average transtation error between ground truth

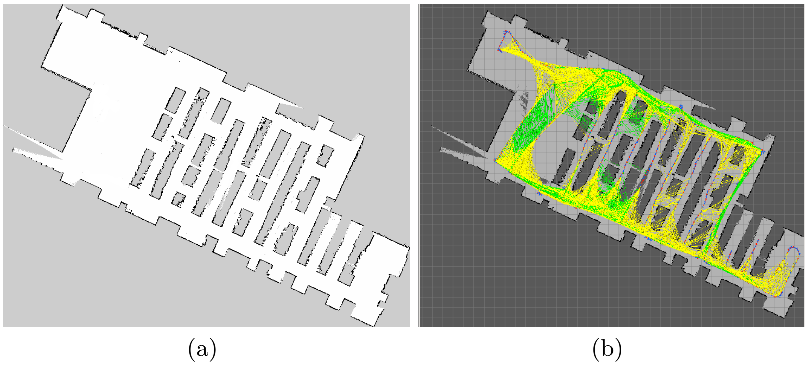


Fig. 5. Mapping results for the data-center room. (a) Resulting map using our approach; (b) Robot trajectory during mapping and graph constraints. Red and blue arrows represent the trajectory; yellow lines represent orientation-only constraints; green lines represent full-pose constraints.

(annotated by hand) and the localization estimate in three different waypoints during the experiment was $0.2m$, while the average angular error was $3.9rad$.

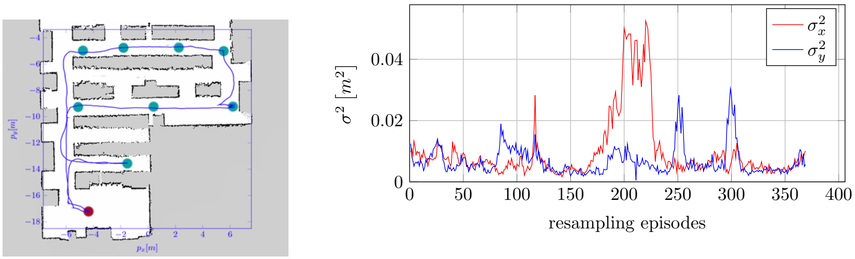
5 Conclusion

In this paper we present an autonomous laser-based mobile robotic system for heat monitoring in data-center rooms. The robot is able to operate in a previously unknown environment, by autonomously creating a map of the environment. Thanks to an ad-hoc web interface, an user is able to set waypoints within the map. The robot is able to localize inside the map, to follow a path by reaching each of the target points and to make environmental measurements. The robot is able to find alternative ways to reach a waypoint if occluded and to avoid moving objects or people. We discuss the practical issues involved with the sensors that we used and the real scenario and showed some preliminary results. Ongoing work is being devoted to exploiting sensor fusion between traditional sensors and vision sensors, as well as to the development of a complete monitoring application.

References

[1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics.* MIT press, 2005.

[2] Fabrizio Abrate et al. “Cooperative robotic teams for supervision and management of large logistic spaces: Methodology and applications”. In: *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on.* IEEE. 2010, pp. 1–8.



(a) (b)

Fig. 6. Localization and path planning performances. (a) Trajectory estimated by localization is shown in blue. Waypoints are shown as green dots, the red dot represents the starting point. (b) Evolution of the variance of the particle cloud during the experiment.

[3] Thomas Ristenpart et al. "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds". In: *Proceedings of the 16th ACM conference on Computer and communications security*. ACM. 2009, pp. 199–212.

[4] Jonathan Koomey. "Growth in data center electricity use 2005 to 2010". In: *The New York Times* 49.3 (2011).

[5] Michael K Patterson. "The effect of data center temperature on energy efficiency". In: *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*. IEEE. 2008, pp. 1167–1174.

[6] Kenneth G Brill. *Data center energy efficiency and productivity*. 2007.

[7] Chandrakant D Patel et al. "Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications". In: *Proceedings of IPACK*. Vol. 1. 2001, pp. 8–13.

[8] Chandrakant D Patel et al. "Smart cooling of data centers". In: *ASME 2003 International Electronic Packaging Technical Conference and Exhibition*. American Society of Mechanical Engineers. 2003, pp. 129–137.

[9] Parthasarathy Ranganathan et al. "Ensemble-level power management for dense blade servers". In: *ACM SIGARCH Computer Architecture News*. Vol. 34. 2. IEEE Computer Society. 2006, pp. 66–77.

[10] Cullen E Bash, Chandrakant D Patel, and Ratnesh K Sharma. "Dynamic thermal management of air cooled data centers". In: *Thermal and Thermomechanical Phenomena in Electronics Systems, 2006. ITHERM'06. The Tenth Intersociety Conference on*. IEEE. 2006, 8–pp.

[11] Ripal Nathuji, Canturk Isci, and Eugene Gorbatov. "Exploiting platform heterogeneity for power efficient data centers". In: *Autonomic Computing, 2007. ICAC'07. Fourth International Conference on*. IEEE. 2007, pp. 5–5.

[12] Rajarshi Das et al. "Autonomic multi-agent management of power and performance in data centers". In: *Proceedings of the 7th international*

joint conference on Autonomous agents and multiagent systems: industrial track. International Foundation for Autonomous Agents and Multiagent Systems. 2008, pp. 107–114.

[13] Luca Parolini, Bruno Sinopoli, and Bruce H Krogh. "Reducing data center energy consumption via coordinated cooling and load management". In: *Proceedings of the 2008 conference on Power aware computing and systems, HotPower*. Vol. 8. 2008, pp. 14–14.

[14] Hendrik F Hamann et al. "Methods and techniques for measuring and improving data center best practices". In: *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*. IEEE. 2008, pp. 1146–1152.

[15] Hendrik F Hamann et al. "Uncovering energy-efficiency opportunities in data centers". In: *IBM Journal of Research and Development* 53.3 (2009), pp. 10–1.

[16] Woong Choi, Ki-Woong Park, and Kyu Ho Park. "SCOUT: Data center monitoring system with multiple mobile robots". In: *Networked Computing and Advanced Information Management (NCM), 2011 7th International Conference on*. IEEE. 2011, pp. 150–155.

[17] Jonathan Lenchner et al. "Towards data center self-diagnosis using a mobile robot". In: *Proceedings of the 8th ACM international conference on Autonomic computing*. ACM. 2011, pp. 81–90.

[18] *ROS (Robot Operating System)*. Website. `http://www.ros.org`.

[19] F. Lu and E. Milios. "Globally consistent range scan alignment for environment mapping". In: *Autonomous Robots* 4 (1997), pp. 333–349.

[20] G. Grisetti, C. Stachniss, and W. Burgard. "Non-linear Constraint Network Optimization for Efficient Map Learning". In: *IEEE Trans. on Intelligent Transportation Systems* 10.3 (2009), pp. 428–439.

[21] R. Kummerle et al. "G2o: A general framework for graph optimization". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. 2011, pp. 3607 –3613. DOI: `10.1109/ICRA.2011.5979949`.

[22] Andrea Censi. "An ICP variant using a point-to-line metric". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Pasadena, CA, 2008. DOI: `10.1109/ROBOT.2008.4543181`.

[23] Luca Carlone et al. "Graph Optimization with Unstructured Covariance: Fast, Accurate, Linear Approximation". In: *Simulation, Modeling, and Programming for Autonomous Robots*. Ed. by Itsuki Noda et al. Vol. 7628. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 261–274. ISBN: 978-3-642-34326-1. DOI: `10.1007/978-3-642-34327-8_25`.

[24] D. Fox. "KLD-Sampling: Adaptive Particle Filters". In: *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[25] I. Ulrich and J. Borenstein. "VFH+: reliable obstacle avoidance for fast mobile robots". In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*. Vol. 2. 1998, 1572–1577 vol.2. DOI: `10.1109/ROBOT.1998.677362`.