Offloading SLAM for Indoor Mobile Robots with Edge-Fog-Cloud Computing

V. K. Sarker[1], J. Peña Queralta[1], T. N. Gia[1], H. Tenhunen[2] and T. Westerlund[1]

[1]Department of Future Technologies, University of Turku, Turku, Finland
[2] Department of Electronics, KTH Royal Institute of Technology, Stockholm, Sweden
Email: [1]{vikasar, jopequ, tunggi, tovewe}@utu.fi, [2]hannu@kth.se

Abstract—Indoor mobile robots are widely used in industrial environments such as large logistic warehouses. They are often in charge of collecting or sorting products. For such robots, computation-intensive operations account for a significant percentage of the total energy consumption and consequently affect battery life. Besides, in order to keep both the power consumption and hardware complexity low, simple micro-controllers or single-board computers are used as onboard local control units. This limits the computational capabilities of robots and consequently their performance. Offloading heavy computation to Cloud servers has been a widely used approach to solve this problem for cases where large amounts of sensor data such as real-time video feeds need to be analyzed. More recently, Fog and Edge computing are being leveraged for offloading tasks such as image processing and complex navigation algorithms involving non-linear mathematical operations. In this paper, we present a system architecture for offloading computationally expensive localization and mapping tasks to smart Edge gateways which use Fog services. We show how Edge computing brings computational capabilities of the Cloud to the robot environment without compromising operational reliability due to connection issues. Furthermore, we analyze the power consumption of a prototype robot vehicle in different modes and show how battery life can be significantly improved by moving the processing of data to the Edge layer.

Index Terms—Edge, Fog, Cloud, SLAM, Efficiency, Computation, Offloading, Energy, Performance, Mobile, Robots

I. Introduction

Small indoor robots which work alongside humans have become ubiquitous in different fields such as home-care, restaurant business and manufacturing plants. For instance, lightweight robots are used in exhibitions for guiding visitors to a desired stall, and are the operational backbone in logistic warehouses of the world's top e-commerce companies. In smart homes, small robots can be used as a pet which can communicate, watch over or play with children and elderly people and can notify or trigger alarm when abnormal situation occurs [1]. Companion robots can remind elderly people to take medicine on time. However, small robots have limited resources such as low battery capacity and light processing capability. In most cases, these robots cannot operate for a long period of time or cannot make useful decisions in complex situations due to these limitations.

Internet of Things (IoT) can be defined as a virtual platform which allows both physical and virtual objects to be interconnected and communicate with each other. It consists of advanced technologies such as sensor fusion, wireless sensor networks and Cloud computing to help improve the quality of services [2]–[6]. Besides, global storage allows users to access data anywhere at any time [7]. Proper utilization of IoT can help overcome some of the drawbacks of small robots. Particularly, real-time position of a small robot can be tracked and a system administrator can send control commands in cases of complex situations [8]. However, IoT-assisted small robots still have some disadvantages. For example, traditional gateways in IoT platform are mainly responsible for collecting data from robots and forwarding the data to Cloud servers. When the data volume is tremendous e.g., due to a large number of multi-sensor equipped robots operating simultaneously, the network bandwidth may be not adequate or the communication latency become too high [9]. Hence, an appropriate architecture is required which can overcome these limitations.

Edge and Fog computing can be described as extra layers between sensor devices, gateways and Cloud servers for enhancing the quality of services. These bring Cloud computing paradigms to the Edge of the network and help reduce the burden of Cloud providing faster services unsupported by Cloud computing. Edge and Fog computing help to reduce energy consumption of sensor nodes and diminish overall latency [10], [11]. The combination of Edge and Fog computing with IoT can provide a suitable approach for enhancing overall performance.

In this paper, we exploit the concept of IoT and Fog along with Edge gateways to enhance energy efficiency and operational performance of small indoor robots. We present a proof of concept for a hybrid Edge-Fog-Cloud architecture which facilitates use of Fog/Edge and Cloud computing for offloading computationally expensive tasks from robots with limited processing power. We discuss related works on how intensive tasks can be offloaded to Edge, Fog or Cloud services in Section II. Section III presents the proposed system architecture applying Edge and Fog computing. Section IV illustrates the results of different tests and discusses the advantages in terms of power saving and performance boost. Section V concludes the work and depicts on future improvements.

II. Related Work

Several works focus on increasing the performance of different algorithms which cannot be fully exploited in resource-

constrained devices. Dey *et al.* analyzed the advantages of leveraging Fog and Edge computing to offload computationally heavy tasks within SLAM algorithms for mobile robots [12]. They provided extensive simulations for different Robot-Edge-Fog-Cloud computational tasks distribution schemes. The authors concluded that Edge and Fog computing can bring significant enhancement to computationally intensive tasks such as localization and positioning tasks. Edge and Fog computing are shown to provide the same computational performance than Cloud offloading, with the benefit of reduced bandwidth and thus faster response time. However, the authors did not perform real-life tests where the communication layer can play an important role and become a bottleneck due to the high data rates between mobile robots and gateways.

Offloading Simultaneous Localization and Mapping (SLAM) tasks to Edge and Fog layers has only been considered recently while Cloud-based SLAM has been studied for the last decade. Benavidez *et al.* deployed an instance of the Robot Operating System in the Cloud to increase the computational capabilities of constraint robots for Visual SLAM (VSLAM) [13]. They pointed how this helps to overcome traditional bottlenecks in robots with limited computational resources with VSLAM performing feature identification and matching which usually need large databases. They demonstrated how parallel computing can be exploited with a multi-node Cloud deployment to facilitate multi-robot SLAM and with limited impact on the on-board computers that control the robots. A similar, more comprehensive work was carried out by Riazuelo *et al.* who presented C2TAM, a Cloud framework for cooperative tracking and mapping [14]. The authors used the unlimited computational resources available in the Cloud to enable the implementation of algorithms which fail to run on single board computers or even more powerful CPUs when the number of agents increases.

Similarly, Turnbull *et al.* utilized Cloud servers to enable advanced neural command for formation control in large groups of robots [15]. However, the network connection between robots and Cloud servers is unreliable and cannot be guaranteed. Therefore, a trade-off exists between the robustness of the solution in terms of available computational resources, operational reliability and performance. This is further explored by Salmeron-Garcia *et al.* in their work on Cloud offloading of a vision-based navigation system [16]. The authors stress in their paper that processing time can be matched to communication period with parallel computation in the Cloud and inferred of a bottleneck due to communication bandwidth and latency. They concluded that Cloud is necessary if the system needs to be easily scalable or precise operation is required. This occurs where the localization or navigation requirements are such that high quality images need to be processed. Extensive works exist in this direction by other authors in the field [17] [18], taking the trade-offs into account both in terms of communication and computational performance.

Kumar *et al.* gathered computational offloading techniques and algorithms into a survey where they accentuate how shifting of intensive applications will be essential in the future for battery powered devices [19] and explored the energy-saving potential of offloading computation to the Cloud.

Extensive work has been carried out on leveraging Cloud computing for offloading heavy computational tasks. However, most of this work have been focused on leveraging virtually unlimited computational resources of the Cloud. Fewer research have targeted the Edge and Fog computing paradigms as a solution to the high latency and unreliability of Cloud services. Keeping these in mind, we focus in both energy efficiency and performance improvement by placing the most expensive computational tasks in Edge and Fog layers leaving the Cloud only for monitoring and management and thus not entirely relying on network link to the Cloud services for normal operation.

III. System Architecture

The proposed system architecture consists of 4 layers: *Robot Layer*, *Edge Layer* for data processing and analysis, *Fog layer* for distributed storage and *Cloud Layer* for monitoring and general mission control. As shown in Fig. 1, we have combined the Edge and Fog layer because the Fog layer consists of the smart Edge Gateways and they use the same physical resources. The architecture we present is easily scalable, modular and distributed by definition. This section thoroughly describes the role of each layer and the distribution of the computational load through the network. Energy consumption, latency and computational power at different layers are the key factors taken into account when deciding the role of a layer.

In general, data is acquired by robots where minimal analysis is performed. Real-time decisions are taken at the Edge gateways to minimize the latency and share the computational load of different connected robots. The Fog layer provides additional services such as distributed storage and processing to enable an efficient handover mechanism with minimal data loss when a robot switches from one gateway to another. The handover includes sharing previous map data acquired by the robot switching gateways so that the result of the SLAM algorithm remains constant through the connection change.

Edge gateways also play an important role in reducing the energy consumption at the end nodes or robots by running the most computational expensive operations. Both Fog and Edge computing are essential in safety critical situations where they provide a more robust situational awareness and overall system control, and robot state knowledge. Finally, the Cloud layer enables end-users or administrators to control the system by giving general instructions and monitor the performance at different layers.

A. Robot Layer

In the robot layer, sensor data is gathered and streamed in real time to the smart Edge gateways. The control of the robot is run online and movement instructions are given in real-time. In the proposed system, the robot can be either aware or unaware of its current state depending on how extensively information is analyzed on-board and whether an
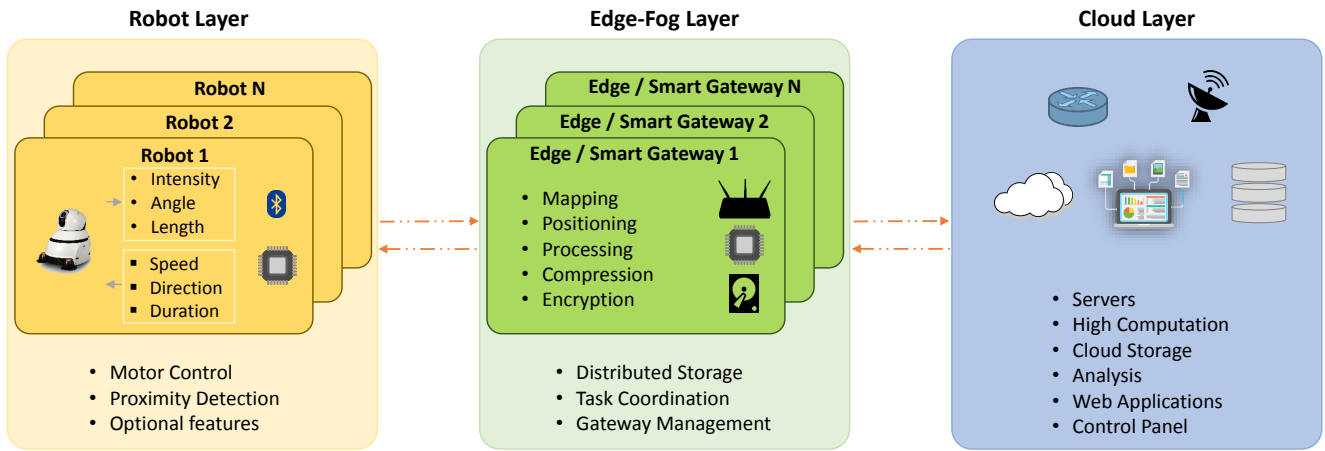
Fig. 1. Proposed System Architecture

Edge gateway provides enhanced state information to the robot or only movement instructions.

The robot relies on low power wireless communication technologies such as Bluetooth or nRF if the bandwidth requirements are met, or Wi-Fi for more bandwidth-intensive applications such as high-quality video transmission. By streaming sensor data directly to the gateway without being processed by the MCU, the robot gains in energy efficiency. Furthermore, low-power techniques can be applied to the MCU in order to further reduce the overall power requirements.

Compared to a traditional approach where sensing, obstacle detection, path planning and localization are made on-board the robot, we can significantly reduce the computing requirements of the robot's control system and downgrade them from processor-based to MCU-based. Only the most basic information is analyzed at the robot itself. This includes, for instance, wheel odometry or inertial data so that the robot is aware of its current acceleration, velocity and orientation. This can be combined with state estimation performed online at the Edge, Fog or Cloud layers to allow more accurate movement and simplify the instructions given to the robot.

B. Smart Edge Gateway Layer

Edge gateways receive sensor data from one or multiple robots in real-time. This includes, for example, images from on-board cameras or data from range sensors such as Lidars and radars, which is analyzed at the gateways to perform obstacle detection and avoidance, path planning, localization or mapping. Odometry data (e.g. wheel odometry, visual odometry) and inertial data (e.g. magnetometer, accelerometer and gyroscope data) is also transmitted from and processed at the gateway.

Acting as a central element, Edge gateways ensure a low-latency and robust solution and provide fast decision making while relocating computationally expensive tasks from the robots. Moreover, if multiple robots are connected to a single gateway, data from different sensors and sources is aggregated and analyzed in order to obtain a more comprehensive

understanding of the environment. For instance, two robots operating in the same environment and connected to the same gateway are able to obtain information of larger areas through sensors of other robots and see through other agents if these are nearby and blocking their field of view.

The role of the Edge layer is of superior importance in safety-critical situations and scenarios such as industrial environments where human share operational space with autonomous or semi-autonomous robots. Compared to the more traditional practice of moving complex tasks to the Cloud, the Edge layer reduces the latency to the point where safe operation is achievable even in the case of network connectivity failure. From an operational point of view, smart Edge gateways receive information from sensors aboard multiple robots and are in charge of decision-making in terms of robot movement and task allocation. These instructions are transmitted continuously and wirelessly to the robots.

C. Fog Layer

The Fog layer includes the smart Edge gateways as part of it. Nonetheless, it is defined as a separate layer because its role in the proposed system architecture differs from the data analysis and instantaneous control role of the Edge layer. The Fog layer represents the interconnection of different Edge gateways, together with other services such as distributed databases or location services. In particular, the Fog layer takes care of the handover mechanisms when the robots disconnect from one gateway and connect to another one together with the procurement of additional services including distributed storage, external tracking and monitoring or location services.

Detailed research has been performed to minimize latency and data-loss during the handover [20] [21]. An alternative for managing the handover is to deploy Edge gateways just behind conventional gateways such as Wi-Fi routers so that the gateways can directly store the received information in distributed storage. Thus, the handover problem would be reduced to the distribution of the computational load of individual robots into

the set of Edge nodes, since all nodes are able to access robot data in real-time from the shared storage.

Localization algorithms are mostly run in the Edge layer in which real-time sensor data is matched with an area of an existing map. This is crucial in situations where the robot operates in a partially or totally unknown environment and simultaneous localization and mapping algorithms are run in the smart Edge gateways. In these cases, it is required that the local map stored in one gateway is shared to other gateways before or during the handover. Accordingly, our proposed system architecture includes a distributed storage deployment in the Fog layer. This can be implemented, for instance, in the form of a decentralized database where the maps and other essential robot data are stored and available to all nodes in the Edge/Fog layers at any time.

Other services included in the Fog layer are external sensor management, collaborative processing and monitoring. For example, while Edge gateways process the on-board sensor information from the robots, external monitoring cameras and other sensors can be used to increase the accuracy of localization algorithms or assure collision avoidance in cases of sensor failure or other local problems that might appear in the robot and go undetected by the Edge gateway. This results in an enhanced situational awareness [22] where both externally and internally collected information about a robot is available in the Fog layer. By introducing additional services in the Fog layer, we can enhance overall system robustness and its fault tolerance. For instance, if a gateway abruptly disconnects or any kind of failure occurs, all the information that it was handling will be available to other gateways due to the shared storage resources. This allows the robot to reconnect to the next available gateway and continue its operation with minimal data loss, operational interruption and latency.

D. Cloud Services

Time-series data of the robot state, including basic information such as position, velocity, acceleration, steering drift factor, drive torque variation, instantaneous current consumption and battery level is uploaded to the Cloud for monitoring, control and visualization by users or administrators [23]. Instead of uploading raw sensor data, it is pre-processed and compressed in the Edge/Fog layers and only sporadic local map updates or critical information is sent. This can be implemented in a dynamic way in which, for instance, the map around a robot or of a given area in the operation environment is updated and uploaded to the Cloud at different frequencies depending on whether an end-user is actively monitoring the operation or not.

The use of Cloud services facilitates generic control instructions to the robot and stable operation [24]. For example, factory managers can easily have an overview of the position of different robots around the factory floor, or override autonomous roaming to involve more robots assigned within a certain group or to prioritize specific tasks. Besides, the Cloud allows global access for the administrators and mission

Fig. 2. Prototype car with Lidar running on top, used for the experiment

planners to have a collaborative overview in a cost-effective way.

However, due to comparatively higher end-to-end latency between the robot layer and the Cloud, it should not be used for primary control purpose as it can be subject to variation in network performance. The Cloud provides access to pertinently processed data from the robot layer which is periodically backed up at a reasonable interval. This yields a conclusive operation and data history which can be analyzed using high performance computing in the Cloud for machine learning, better prediction and big data analytics.

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

In order to test the feasibility and validate the proposed architecture, we have constructed a car controlled via Bluetooth communication to produce a map of the perimeter it is bounded by. The car shown in Figure 2 incorporates a 360° Lidar on top of the chassis, two independent motors- one for steering control and another for driving forward or backward with appropriate motor control circuitry. The car is connected to a gateway through a Bluetooth Classic module. An AVR 8-bit MCU is used to control acquisition of raw data from the Lidar and send to gateway via the serial port profile of the Bluetooth standard. A prototype is built using the chassis from an *Reely 1:10 Elektro-Monstertruck NEW1* RC car and replacing the control circuit and motors with lower power variants with an RPLIDAR A1 M8 on the top, a low cost 360° 2D LASER scanner from *Slamtec* [25].

We have run experimental tests with a minimal setup where only Lidar is used for both localization and mapping purposes. Adding inertial sensors would provide more accurate positioning and control over the vehicle; however, that is not the main objective of this work and we focus on the implementation of the hybrid Edge-Fog-Cloud architecture and its benefits.

The MCU controlling the movement of the car does not save or analyze the data from the Lidar sensor in any form and is forwarded directly to the gateway as serial stream over Bluetooth. The gateway then runs an adapted version of the BreezySLAM algorithm by S. D. Levy [26]. The algorithm has been modified to take into account the intermittent nature of
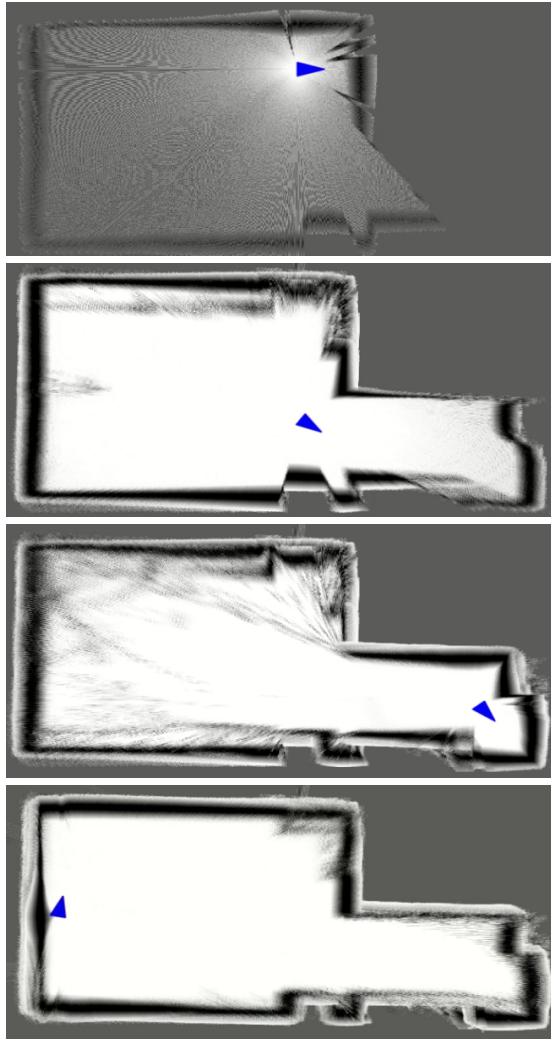
Fig. 3. Mapping and Localization at the Edge. (The position of the car is marked with a blue pointer)

Operation / Approach	Average Current (mA)	Average Power (mW)
MCU Idle	4	19
MCU + Lidar	339	1695
MCU + Lidar + Streaming	378	1890
R.Pi Idle	301	1503
R.Pi + Lidar	622	3108
R.Pi + Lidar + Edge Processing	829	4152
Car (Drive / Steer)	122	607
Car (Drive + Steer)	184	918

the connection between the robot and the gateway, and execute accordingly when processing successive batches of data.

Figure 3 shows the result of applying the mapping and localization algorithm in the Edge gateway. As we are using an inexpensive Lidar and the SLAM algorithms run without any inertial measurement unit such as accelerometer and magnetometer, accurate positioning is hard to achieve in complex environments. However, with more input data from multiple sensors would suffice for enabling operation in non-trivial local environments.

The first map in Figure 3 shows the map obtained after the first few Lidar frames have been analyzed. The *RPLIDAR A1 M8* used in the tests is a low-cost Lidar and therefore several scans are processed before obtaining a good quality map. The granularity in the first map is derived from the relatively low scan speed of the device compared to more advanced sensors. Also, while the position within the map is accurate and consistent through the test, noise appears near the distant walls from the Lidar due to the low accuracy. However,

the walls and corners in the local area around the robot are with little to no noise.

In terms of performance, the recorded path elapsed a total of 44.3 seconds in which 314 Lidar frames were processed. This translates into 7 frames/s, which matches with the Lidar specification for a maximum speed scan of 10 Hz with an average car speed of 0.3 m/s. This frame rate can be compared to 1 frame/s processed on a Raspberry Pi 3 Model B running Ubuntu Mate Desktop with the graphical map display in real-time. While this can be optimized running a more trimmed, customized version of Linux on the Raspberry Pi, its computational capability limits the processing to about 5 frames/s. Since a MCU is able to handle the vehicle control and transmission of Lidar data, not only it significantly increases the system performance by offloading the SLAM tasks to the Edge gateway, but a considerable amount of energy is saved by replacing a power-hungry single-board computer such as the Raspberry Pi with a low-power MCU.

In particular, the specific power consumption and average current drawn by the prototype in different states are documented in Table I. In idle state, the Lidar consumes 300 to 350 mA, representing over 90% of the total power consumption. This can be replaced with low-power cameras or more energy efficient sensors for a particular SLAM implementation. However, when the prototype is moving, the Lidar's share of the power consumption goes down to around 60%. When choosing the Lidar, general purpose motors have been used to show that even with out-of-the-shelf devices, the overall energy efficiency can be improved by around 40% when a single-board computer such as a Raspberry Pi in this test is replaced with a MCU. This yields a significantly longer battery life and an increase in operational performance due to larger ratio of active-time with respect to idle or charging time.

V. DISCUSSION

Although Edge computing can bring advantages such as reduction in latency, bandwidth conservation, improvement in application robustness and security [11], [27], [28], there are inherent challenges. As the processing of the data stays near the Edge, specific data-oriented applications which require comparatively higher user interaction will have insignificant

performance gain. In the worst case scenario, performance can deteriorate due to the long path data has to travel.

As multiple robots can connect to the same smart gateway, communication latency and inter-unit interference can be an issue. Depending on the application, an appropriate communication medium must be chosen to ensure that specific parameters such as bandwidth, channel spacing, protocol, sampling frequency and data rate are properly considered for an interruption-free data transfer. Also, for wireless units, it is useful to examine the wake-up and sleep time of the radio communication module so that transmitter and receiver circuitry is not toggled unnecessarily and sleep modes are used appropriately. This yields better energy efficiency and communication at an optimal duty cycle.

Keeping the Edge gateways near the robot layer and a Fog layer provides distributed and scalable computing. However, as the number of robots and frequency of data transmission per node significantly increases, the gateways may not handle all requests within the highest permitted latency. In such a case, having more powerful and redundant gateways along with advanced load-sharing algorithm in the Fog and distributed data storage can effectively improve performance.

VI. CONCLUSION AND FUTURE WORK

In this paper, the comparative advantage of using Edge computing for robots and nodes are discussed keeping the focus on energy efficiency and operational latency. Instead of directly transferring a large amount of data from robots to the Cloud, we proposed an Edge-Fog-Cloud based system for performance advantages. Run from the mains supply, the powerful Edge layer can pre-process and analyze data easily, implement advanced features and ensure data security by applying complex encryption algorithms. The Fog layer can intelligently manage the smart gateways of Edge layer.

This approach effectively takes the heavy computational tasks near the Edge layer and thus reduces the required energy for operation of the node or robot itself. The shorter path of data for a complete cycle improves latency and frees up valuable network bandwidth, especially when a lot of robots are connected to the same gateway. In addition, the reduced robot-to-Cloud transferable data eliminates potential bottleneck in the network and lowers the chance of system downtime due to a single point computing failure as in centralized computing architecture. We demonstrated an example of SLAM showing how a car in the robot layer can take advantage of the power of Edge computing for mapping a confined perimeter.

In future, we plan to implement a comprehensive system consisting of several vehicles, multiple smart Edge computing gateways constituting a Fog layer and an application for the Cloud layer. In addition, we plan to develop an autonomous roaming algorithm for the robot car so as to perform the mapping operation in shortest possible time. Besides, some stress tests to estimate the performance of smart gateways will be evaluated to emulate the scenario when the number of connected cars is sufficiently high.

REFERENCES

[1] N. Mitabe and N. Shinomiya. An intelligent care support system for the elderly with an autonomous robot in ambient sensors. In *2017 IEEE GCCE*, pages 1–4, 2017.

[2] M. Jiang *et al.* Iot-based remote facial expression monitoring system with semg signal. In *2016 IEEE SAS*, pages 1–6. IEEE, 2016.

[3] S. R. Moosavi *et al.* Session resumption-based end-to-end security for healthcare internet-of-things. In *IEEE CIT*, pages 581–588. IEEE, 2015.

[4] M. Ali *et al.* Autonomous patient/home health monitoring powered by energy harvesting. In *IEEE GLOBECOM 2017*, pages 1–7. IEEE, 2017.

[5] T. N. Gia *et al.* Iot-based continuous glucose monitoring system: A feasibility study. *Procedia Computer Science*, 109:327–334, 2017.

[6] I. Tcarenko *et al.* Energy-efficient iot-enabled fall detection system with messenger-based notification. In *International Conference on Wireless Mobile Communication and Healthcare*, pages 19–26. Springer, 2016.

[7] R. Petrolo *et al.* Integrating wireless sensor networks within a city cloud. In *IEEE SECON Workshops*, pages 24–27, 2014.

[8] T. Kitanouma *et al.* Cloud-based self-organizing localization with virtual network topology for wireless sensor networks and its implementation. In *2016 IEEE PIMRC*, pages 1–7, 2016.

[9] W. Yu *et al.* A survey on the edge computing for the internet of things. *IEEE access*, 6:6900–6919, 2018.

[10] B. Negash *et al.* Leveraging fog computing for healthcare iot. In *Fog Computing in the Internet of Things*, pages 145–169. Springer, 2018.

[11] T. N. Gia *et al.* Exploiting fog computing in health monitoring. *Fog and Edge Computing: Principles and Paradigms*, pages 291–318, 2019.

[12] S. Dey and A. Mukherjee. Robotic slam: a review from fog computing and mobile edge computing perspective. In *MobiQuitous*, pages 153–158. ACM, 2016.

[13] P. Benavidez *et al.* Cloud-based realtime robotic visual slam. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings*, pages 773–777, 2015.

[14] L. Riazuelo *et al.* C2tam: A cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, 62(4):401 – 413, 2014.

[15] L. Turnbull and B. Samanta. Cloud robotics: Formation control of a multi robot system utilizing cloud infrastructure. In *2013 Proceedings of IEEE Southeastcon*, pages 1–4, 2013.

[16] J. Salmerón-Garcı´a *et al.* A tradeoff analysis of a cloud-based robot navigation assistant using stereo image processing. *IEEE Transactions on Automation Science and Engineering*, 12(2):444–454, 2015.

[17] Y. Hong *et al.* Energy efficient content-based image retrieval for mobile systems. In *2009 IEEE International Symposium on Circuits and Systems*, pages 1673–1676, 2009.

[18] J. Hauswald *et al.* A hybrid approach to offloading mobile image classification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8375–8379, 2014.

[19] K. Kumar *et al.* A survey of computation offloading for mobile systems. *Mobile Networks and Applications*, 18(1):129–140, 2013.

[20] X. Wang *et al.* Offloading in internet of vehicles: A fog-enabled real-time traffic management system. *IEEE Transactions on Industrial Informatics*, 14(10):4568–4578, 2018.

[21] T. N. Gia *et al.* Fog computing approach for mobility support in internet-of-things systems. *IEEE Access*, 6:36064–36082, 2018.

[22] C. V. Smith *et al.* Enhanced situational awareness in autonomous mobile robots using context-based mapping. In *2013 IEEE CogSIMA*, pages 134–138, 2013.

[23] B. Zhou *et al.* Remote control system of mobile robot based on cloud platform. In *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, pages 94–98, 2017.

[24] C. Briese *et al.* Cloud-based active disturbance rejection control for industrial robots. In *2018 IEEE ETFA*, pages 559–565, 2018.

[25] Slamtec. RPLIDAR A1. [Online] Available: http://www.slamtec.com/en/lidar/a1. Accessed: 20 Feb. 2019.

[26] S. D. Levy. Breezyslam. [Online] Available: https://github.com/simondlevy/BreezySLAM. Accessed: 20 Feb. 2019.

[27] T. N. Gia *et al.* Fog computing in healthcare internet of things: A case study on ecg feature extraction. In *2015 IEEE CIT*, pages 356–363. IEEE, 2015.

[28] N. Abbas *et al.* Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, 2018.