Offloading Monocular Visual Odometry with Edge Computing: Optimizing Image Quality in Multi-Robot Systems

Li Qingqing
qingqli@utu.fi
University of Turku
Turku, Finland

Jorge Peña Queralta
jopequ@utu.fi
University of Turku
Turku, Finland

Tuan Nguyen Gia
tunggi@utu.fi
University of Turku
Turku, Finland

Tomi Westerlund
tovewe@utu.fi
University of Turku
Turku, Finland

ABSTRACT

Fleets of autonomous mobile robots are becoming ubiquitous in industrial environments such as logistic warehouses. This ubiquity has led in the Internet of Things field towards more distributed network architectures, which have crystallized under the rising edge and fog computing paradigms. In this paper, we propose the combination of an edge computing approach with computational offloading for mobile robot navigation. As smaller and relatively simpler robots become more capable, their penetration in different domains rises. These large multi-robot systems are often characterized by constrained computational and sensing resources. An efficient computational offloading scheme has the potential to bring multiple operational enhancements. However, with the most cost-effective autonomous navigation method being visual-inertial odometry, streaming high-quality images can induce latency increments with a consequent negative impact on operational performance. In this paper, we analyze the impact that image quality and compression have on the state-of-the-art on visual inertial odometry. Our results indicate that over one order of magnitude in image size and network bandwidth can be reduced without compromising the accuracy of the odometry methods even in challenging environments.This opens the door to further optimization by dynamically assessing the trade-off between image quality, network load, latency and performance of the visual-inertial odometry and localization accuracy.

CCS CONCEPTS

• **Computing methodologies → Vision for robotics**; *Tracking*.

ICSCC, December 21-23, 2019, Wuhan, China

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

KEYWORDS

Visual Odometry; Visual-Inertial Odometry; Monocular Visual Odometry; Multi Robot Systems; Edge Computing; Computational Offloading; Internet of Robots; Internet of Vehicles; Image Compression; Image Quality

1 INTRODUCTION

Accurate localization and mapping are two of the pillars behind fully autonomous systems [1, 2]. Over the past two decades, much attention has been put into solving the simultaneous localization and mapping (SLAM) problem [3–5]. What has been a mostly offline or offloaded method due to its computational complexity is now a widely implemented real-time algorithm that runs on on-board computers in mobile robots [6]. Among the different sensors that can provide motion estimation, a visual-inertial system can produce one of the best price-accuracy ratios [7], with cameras and inertial measurement units having prices of several orders of magnitude lower than 3D lidars [8].

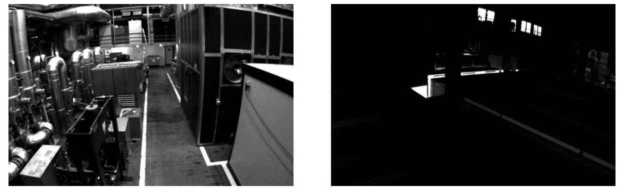
In recent years, research into visual-inertial odometry (VIO) as part of the SLAM problem has attracted increasing interest due to the low price and ease for cross-platform implementation, among other benefits [5]. Visual-inertial odometry has potential for multiple applications, including augmented reality (AR) [9, 10], aerial robotic navigation. The current state of the art can achieve very high accuracy even in a dynamic and challenging environment, both for monocular [11] and stereo vision [12]. Mature algorithms such as RAVIO [13] or VINS-Mono [14] have raised the level of autonomy of drones and small robots with existing hardware, and multiple open-source datasets such as EuroC has been published, pushing the research in this area forward [15].

While visual-inertial odometry enables low-cost and accurate autonomous operation for small mobile robots, it still requires robots to have a minimum of computational resources available on their on-board computers. Most of the current research efforts are focused on algorithmic level optimization to achieve higher levels of accuracy and reliability in visual odometry on different hardware platforms. This has led to high-accuracy methods enabling long-term autonomy with efficient loop closure mechanisms [14]. However, small units such as flying robots usually have constrained resources, including limited power and computational capabilities or reduced storage. In this situation, an aspect to consider is how to reduce the robots' computational burden while maintaining the VIO algorithm's high performance. If multiple cameras are utilized to reduce the blind angles for obstacle avoidance, path planning, and mapping, then the computational burden can increase considerably. This can have a significant impact on the performance and ability to autonomously navigate a complex environment in small mobile robots, including aerial drones. If additionally, multiple robots are operating in the same environment, accurate localization is essential to secure their operation and avoid collisions. In a multi-robot system where robots have equivalent sensing capabilities, the offloading part of the data processing can be a solution that not only increases the reliability of the system but also reduces the unit cost of each robot as the hardware can be simplified. In an industrial environment with large numbers of autonomous robots operating within a controlled area, reducing the cost of each robot can have a direct impact on the industrial ecosystem as a whole.

In recent years, some researchers have introduced the cloud robotics concept, in which the capabilities of small mobile robots can be enhanced by moving part or most of the computationally intensive data analysis tasks to a cloud environment [16, 17]. Nonetheless, streaming data to the cloud has the potential to significantly reduce the overall system reliability with uncontrolled latency or unstable network connection [18, 19]. We extend the recent trend in the IoT towards more decentralized network architectures with the fog and edge computing paradigms [20–22]. Edge computing crystallizes the idea of keeping the data processing as close as possible to where the data originates. With this approach, raw data is processed at the local network level instead of the cloud, decreasing the latency and optimizing the network load [23]. Furthermore, savings in hardware platforms and overall power consumption can be optimized with proper integration of edge computing [24]. In this work, we have moved the VIO computation towards a smart edge gateway to open the possibility for more intelligent, yet simple, large teams of autonomous robots that rely on edge services for offloading most of their computationally intensive operation.

The main motivation behind this paper is to study the optimal relationship between image quality and accuracy of a monocular visual odometry algorithm in a computational offloading scheme. Finding the proper trade-off between accuracy and image size has a direct impact on the computational resource consumption, algorithm runtime, network latency and, in consequence, the number of robots that can be supported simultaneously from a single smart edge gateway. Our goal is to provide a benchmark of the compression rate's influence on the VIO algorithm. To address these issues, we employ the state of art VIO algorithm VINS-Mono [14] and analyze its performance on an open dataset, the EuRoC MAV dataset [15], with varying image compression rate and picture quality. Our results show that the computational offloading scheme can be optimized in terms of bandwidth usage without compromising the accuracy of the visual odometry algorithm. Furthermore, decreasing the image quality reduces the processing time at the edge gateway. Therefore, finding the appropriate compression rate not only optimizes the network load but also enables a single gateway to handle the odometry for a larger number of connected robots.

The main contribution of this paper is on analyzing the performance of the state-of-the-art in monocular visual odometry with varying image quality and compression settings. We utilize the



(a) (b)

Figure 1: EuRoC dataset samples. Subfigure (a) shows a sample of the easier environment for odometry, while (b) shows the harder dataset.

JPEG standard and examine the performance of a monocular visual odometry algorithm with the JPEG image compression setting varying from 1% to 100%. The implications of this study can be significant in a computational offloading scheme; an image size reduction of up to two orders of magnitude can be achieved without a significant compromise on odometry accuracy.

The remainder of this paper is organized as follows. In Section 2, we overview related works utilizing computational offloading for visual odometry in mobile robots, mostly with a cloud-based approach. In Section 3, we introduce the basic concepts behind visual odometry, as well as specific algorithms utilized in this paper (VINS-Mono). Section 4 then introduces the methodologies, experimental setup and results, which provide insight towards the optimal image quality to be chosen to minimize network.At last.Section 5, conclude a conclusion and discuss the possible future work.

2 RELATED WORK

The problem of SLAM has been traditionally considered either as an offline problem, where all accumulated data is utilized to rebuild the path, or an online problem for real-time image analysis with an on-board computer. However, if a large fleet of robots is considered, then a computational offloading scheme can considerably bring the cost down. To the best of our knowledge, computational offloading has been considered for mobile robot navigation a mapping only from the cloud computing point of view with cloud-centric architectures and data processing in powerful servers where the algorithms can be easily run in parallel at maximum efficiency. Yun *et al.* proposed a robotics platform to be deployed in cloud servers, RSE-PF, for distribution visual SLAM where data from different robots was aggregated and combined in the cloud [16]. An average network latency of approximately 150 ms was reported (round trip). Even with almost instantaneous data processing at the cloud servers, this either limits the image analysis rate to around 6 frames/second or induces a delay when parallel RX/TX channels are utilized. In the first case, an on-board computer such as a Raspberry Pi 4 or an NVIDIA TX2 could be able to provide a similar o better frame rate, while in the second case an accurate estimation of network latency must be available at the robot in order to interpret properly the processed information that the cloud servers return. The maximum number of robotic units that could be supported simultaneously was not reported; however, the authors utilized WebSockets in order to save bandwidth compared to HTTP. Dey *et al.* proposed a similar offloading scheme in which a multi-tier edge+cloud architecture was introduced [17]. Rather than concentrating on analyzing the

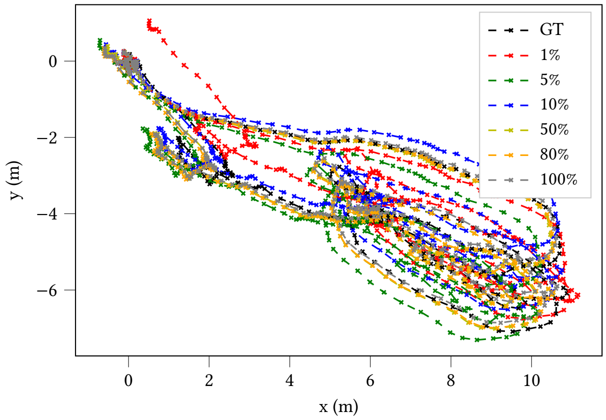


Figure 2: Ground truth and odometry reconstructed paths with the easier dataset.

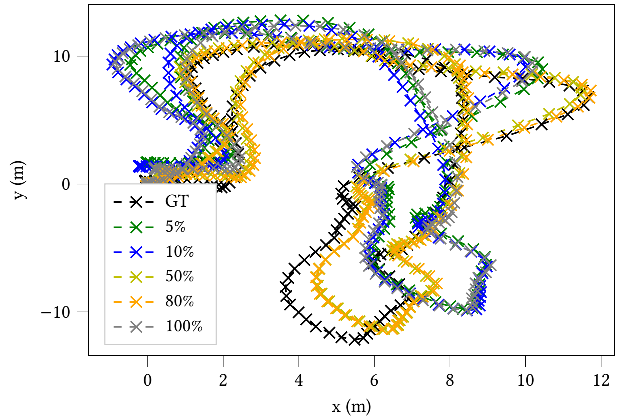


Figure 3: Ground truth and odometry reconstructed paths with the easier dataset.

performance, the authors shifted the research focus towards defining and solving an optimization problem in order to maximize the performance of the multi-tier architecture by offloading different processes to different layers. Their approach was to utilize integer linear programming for optimization of offloading design decisions utilizing the network bandwidth as a variable and adding latency constraints.

In this paper, we extend our previous work in progress report where we analyzed the effect of image compression on the performance of visual odometry [25]. In contrast with the cloud-centric approach that can be found in the literature, we propose the offloading at the local network level following the main design ideas of edge computing. With this method, we are able to keep the benefits of cloud-based offloading (optimization of energy consumption and simplification of on-board hardware) while reducing the latency and increasing the network reliability due to a single connection being used and allowing for more tight bandwidth and network management control. We focus on finding the right trade-off between odometry accuracy and performance in terms of frame rate and latency.

3 MONOCULAR VISUAL INERTIAL ODOMETRY

Visual Odometry(VO) is a part of Visual SLAM (VSLAM) .VO focuses on the local consistency of the robot movement trajectory, using real-time data to predict robot egomotion. The goal of SLAM is to achieve global consistency between odometry and maps. So VO can be used as a building block for VSLAM, before tracking all the camera's historical data to detect loop closure and optimize the map.

3.1 The SLAM problem

SLAM is an abbreviation for Simultaneous Localization And Mapping. SLAM was a term first utilized in the field of robotics but has been applied in many other fields afterward, mostly involding computer vision, virtual reality or augmented reality. It enables robots to construct a map of the surrounding environment in real time based on sensor data without any prior knowledge, and to speculate on its own relative location based on this map.

3.2 Visual Odometry: Monocular VS Binocular

Visual-Inertial Odometry (VIO) is an algorithm that combines camera and IMU data to implement SLAM or state estimation. The advantage of binocular VO is that it can accurately estimate the motion trajectory and is able to recover the exact physical units. In Monocular VO, it is only possible to obtain information regarding what the object has moved as a certain number of relative units in a given direction, while the binocular VO is able to map these relative units to a metric system representing the real length or size. However, for objects that are far away, the binocular system degenerates into a monocular system. Monocular visual odometry has gained increasing attention in recent years because of the lower price and ease of automatic calibration. However, the data processing is more challenging.

3.3 VINS-Mono

VINS-Mono adopts a non-linear optimization-based sliding window estimator to predict a robot's position and orientation. This approach begins with the measurement preprocessing which will collect sensor data to detect feature and IMU pre-integration. Through the initialization procedure, all values for bootstrapping the subsequent nonlinear optimization-based VIO will be calculated. The VIO with relocalization modules tightly fuse integrated IMU measurement processing, feature observation, and redetected features from a loop closure scheme. Finally, the pose graph module implements global optimization to reduce drift.

4 EXPERIMENT AND RESULTS

We have utilized an open-source dataset, the EuRoC dataset, in order to evaluate how the performance of the VINS-Mono algorithm varies when the image quality is reduced [15]. This is an initial approach and we have utilized the standard JPEG compression algorithm since it provides a high range of possible compression rates through its image quality parameter. For instance, given a

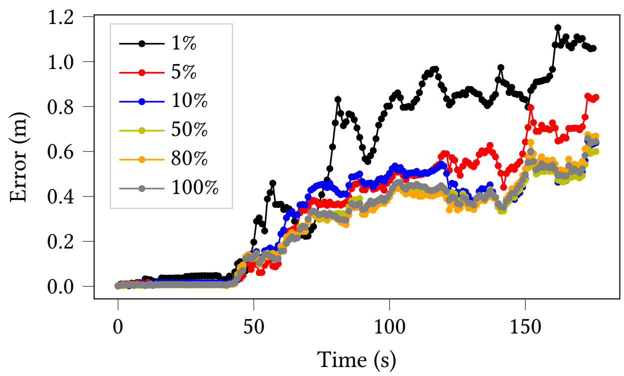


Figure 4: VINS-Mono error in the easier dataset.

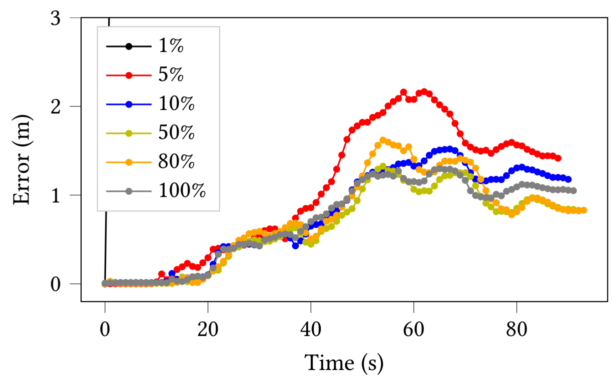


Figure 5: VINS-Mono error in the harder dataset.

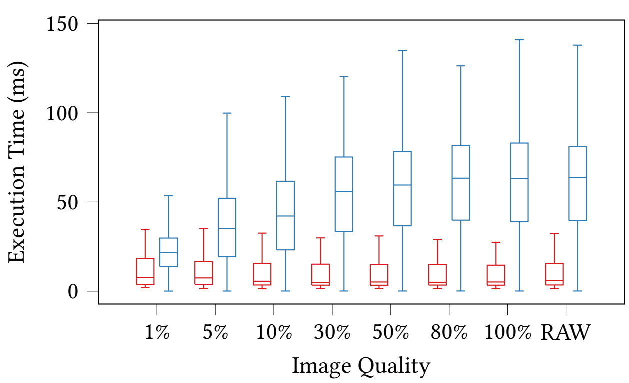


Figure 6: Execution times: feature extraction (red) and pose estimation (blue).

sample from the EuRoC dataset that has a size of 362 kB in PNG format, its size in JPEG ranges from 6.7 kB with 1% quality and 226 kB for 100% quality setting.

The EuRoC dataset is a binocular + IMU dataset for indoor micro aerial vehicles (MAV). It contains two scenes, one is a machine hall, and the other is a normal room. The dataset uses the flying robot AscTec Firefly as a data acquisition platform. It is equipped with

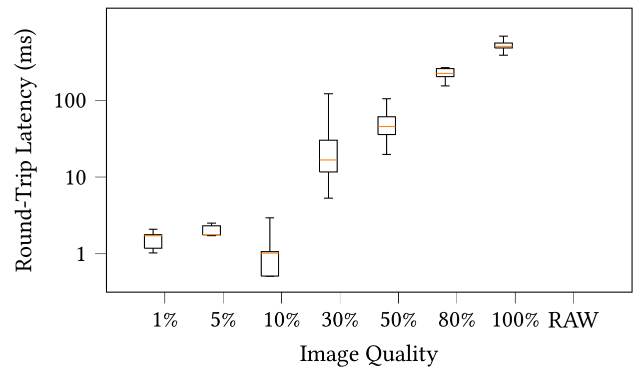


Figure 7: Average round trip latency with a UDP server.

Table 1: Execution time of the different processes and network latency for a subset of image qualities.

	Image Quality				
	1%	**5%**	**10%**	**50%**	**100%**
Image size (kB)	5.7	7.9	11.2	28.3	202.2
Network latency (ms)	1.99	2.11	1.35	77.81	545.71
Feature extraction (ms)	11.603	10.669	9.786	9.124	8.890
Pose Estimation (ms)	23.074	37.261	44.792	58.553	61.105

binocular camera MT9V034 and an IMU ADIS16448. The camera frame rate frequency is 20hz, and the IMU frequency is 200hz. The authors utilize a Vicon motion capture system and Leica Nova MS50 as ground-truth for benchmarking odometry algorithms. Due to the stable and reliable data provided, it has currently become a popular dataset [26, 27].

Our experiments have focused on the analysis of two parameters: the latency of the network and the accuracy of the odometry algorithm. We have also analyzed the processing time required for the feature extraction process and the pose estimation process for each of the image compression ratios. We have utilized two subsets of the EuRoC dataset which are considered easy and hard for visual inertial odometry algorithms, due to the extraction of less or more features. Samples from these two subsets are shown in Figure 1, where it can be seen that the image corresponding to the harder set is much darker and less features can be consequently detected. In fact, in this case, even if an image compression ratio of 5% has an impact of around 25% of the error at the end of the sample path (0.8 m error with 5% quality versus 0.65 m with 100% quality), and 1% quality renders a final error of around 1.1 m. In the harder dataset, however, only up to 5% image quality allows for a convergent path, as with 1% quality the algorithm is unable to calibrate the camera and IMU and the path diverges from the start. The errors accumulated with the VINS-Mono odometry algorithm over the easier and harder paths are shown in Figures 4 and 3, respectively. These indicate that the data quality can be reduced to as little as 10% without compromising the performance, while 50% quality gives the best performance in a harder environment. In the easier case, a 10% quality image matches the best performance with minimal odometry error while achieving two orders of magnitude of reduction in the network latency with respect to broadcasting a raw image.

The two main processes in which an odometry algorithm can be divided are feature extraction and pose estimation. The distribution of the execution times of these processes for a range of image qualities (1% to 100%) is shown in the boxplot in Figure 6, which have been obtained utilizing a 64-bit Intel Core i7-4710MQ CPU with 8 cores at 2.50 GHz. Each of the distributions has been calculated with 1000 images for which the different compression rates have been applied. While the feature extraction process has an execution time that remains constant with the increasing image quality, the pose estimation increases as more features are found in higher quality images. The network latency has an overhead effect that varies from under 1% (image qualities under 10%) to over 700% (100% image quality) when compared to the data processing time (feature extraction and pose estimation). The distribution of round-trip latency for a subset of image qualities is shown in the boxplot in Figure 7, where samples of 100 images have been utilized to calculate each of the distributions.

5 CONCLUSION AND FUTURE WORK

We have evaluated the impact of image compression and quality in a visual inertial odometry algorithm. Our results show that image quality can be reduced up to a certain threshold, which depends on the ability of the algorithm to extract features from the environment, without a significant impact on odometry accuracy. This opens the door to the utilization of an efficient computational offloading scheme with edge computing. In turn, this enables the simplification of hardware onboard robots, a consequent reduction of power consumption and the ability to utilize a single edge gateway to offload the odometry computation from multiple robots. The latency of the network adds an overhead between 0.3% and 780% with respect to the processing time. In both datasets considered, a low accuracy loss could be achieved reducing the image quality to as much as 10%, where the network overhead is below 1%. In consequence, the offloading scheme does not induce significant delays to the odometry and has the potential to even improve the performance in terms of frame rate with more powerful edge gateways. The proposed edge computing offloading scheme can bring multiple benefits to a large multi-robot system, from cost reduction and energy efficiency to increased performance and reliability.

In future work, we will evaluate a wider range of odometry algorithms and image compression methods. We will also compare the execution time of the odometry algorithms on typical on-board computers utilized in aerial robots with multiple instances of the same algorithm running in parallel giving support to multiple robots.

REFERENCES

[1] G. Bresson*et al.* Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Trans. on Intelligent Vehicles*, 2017.

[2] L. Qingqing *et al.* Multi Sensor Fusion for Navigation and Mapping in Autonomous Vehicles: Accurate Localization in Urban Environments. In *The 9th IEEE CIS-RAM*, 2019.

[3] T. Durrant *et al.* Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[4] T. Bailey *et al.* Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006.

[5] J. Fuentes-Pacheco *et al.* Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[6] W. Hess*et al.* Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278. IEEE, 2016.

[7] Sherif A.S. Mohamed *et al.* A survey on odometry for autonomous navigation systems. *IEEE Access*, 2019.

[8] J. Zhang*et al.* Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015.

[9] T. Oskiper*et al.* Camslam: Vision aided inertial tracking and mapping framework for large scale ar applications. In *IEEE ISMAR*, pages 216–217. IEEE, 2017.

[10] S. Cortés*et al.* Advio: An authentic dataset for visual-inertial odometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 419–434, 2018.

[11] A. Hardt-Stremayr*et al.* Towards fully dense direct filter-based monocular visual-inertial odometry. In *IEEE ICRA*. IEEE, 2019.

[12] K. Sun *et al.* Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, 3(2):965–972, 2018.

[13] M. Bloesch*et al.* Robust visual inertial odometry using a direct ekf-based approach. In *IEEE/RSJ IROS*. IEEE, 2015.

[14] T. Qin *et al.* Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[15] M. Burri *et al.* The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[16] P. Yun *et al.* Towards a cloud robotics platform for distributed visual slam. In *Computer Vision Systems*. Springer, 2017.

[17] S. Dey *et al.* Robotic slam: A review from fog computing and mobile edge computing perspective. In *MOBIQUITOUS*. ACM, 2016.

[18] L. Qingqing *et al.* Edge Computing for Mobile Robots: Multi-Robot Feature-Based Lidar Odometry with FPGAs. In *The 12th ICMU*, 2019.

[19] V. K. Sarker *et al.* Offloading slam for indoor mobile robots with edge-fog-cloud computing. In *1st ICASERT*, 2019.

[20] A. Metwaly*et al.* Edge computing with embedded ai: Thermal image analysis for occupancy estimation in intelligent buildings. In *INTelligent Embedded Systems Architectures and Applications, INTESA@ESWEEK 2019*. ACM, 2019.

[21] T. N. Gia *et al.* Artificial Intelligence at the Edge in the Blockchain of Things. In *8th EAI Mobihealth*, 2019.

[22] A. Nawaz *et al.* Edge AI and Blockchain for Privacy-Critical and Data-Sensitive Applications. In *The 12th ICMU*, 2019.

[23] T. N. Gia *et al.* Edge AI in Smart Farming IoT: CNNs at the Edge and Fog Computing with LoRa. In *2019 IEEE AFRICON*, 2019.

[24] J. Peña Queralta *et al.* Edge-AI in LoRa based healthcare monitoring: A case study on fall detection system with LSTM Recurrent Neural Networks. In *2019 42nd International Conference on Telecommunications, Signal Processing (TSP)*, 2019.

[25] L. Qingqing *et al.* Visual Odometry Offloading in Internet of Vehicles with Compression at the Edge of the Network. In *The 12th ICMU*, 2019.

[26] C. Cadena *et al.* Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. on robotics*, 2016.

[27] Raúl *et al.* Mur-Artal. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.