Poster: Enabling Environment-Aware And Task-Oriented Communication for Low-Latency Edge SLAM

Yao Zhang
yaozh.g@nwpu.edu.cn
Northwestern Polytechnical University
Xi'an, China

ABSTRACT

Visual simultaneous localization and mapping (vSLAM) is a prevailing technology for many emerging robotic applications. However, real-time SLAM on mobile robotic systems with limited computational resources is hard to achieve because the complexity of vSLAM algorithms increases over time. This restriction can be lifted by offloading computation to edge servers that have abundant computational resources, forming the emerging paradigm of *edge-assisted SLAM*. Nevertheless, sending high-dimensional vision data over volatile wireless channels inevitably leads to excessive delay, which decelerates map updating and thus degrades the localization accuracy. In this paper, we design a new system architecture that enables low-latency communication for edge-assisted SLAM with improved localization performance. Extensive experiments prove that our system is cost-effective and outperforms better.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

ACM Reference Format:
Yao Zhang. 2024. Poster: Enabling Environment-Aware And Task-Oriented Communication for Low-Latency Edge SLAM. In *Proceedings of CSCAIoTSys: Coupling of Sensing Computing in AIoT Systems, May 13, 2024, HongKong, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The recent advancements in sensing, communication, control, and artificial intelligence (AI), are accelerating the ubiquitous applications of intelligent robotic systems, such as smart manipulators, surgical robots, autonomous vehicles, and drones [2]. To accomplish the complex robotic tasks, autonomous navigation in unknown environments plays a pivotal role, which requires to construct a spatial map of the environment, and meanwhile, locate the robot in the constructed map [1]. Visual SLAM (vSLAM) has been recognized as a prevailing autonomous navigation technology since it facilitates seamless indoor-outdoor navigation and admits centimetre-level accuracy.

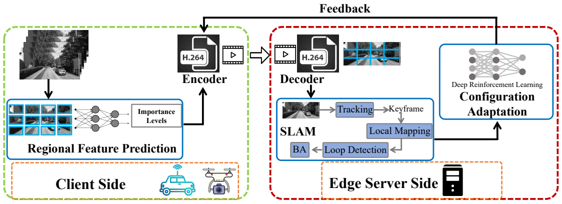

Figure 1: Edge-assisted SLAM with task-oriented communication.

However, the complexity of vSLAM algorithms typically increases as the constructed map grows, which makes it very challenging to achieve real-time operation on mobile robotic systems due to their limited computational resources.

An effective approach is edge SLAM, which splits the SLAM processing pipeline into a client and a server module [4]. The client module is responsible for detecting keyframe features and identifying the keyframes, which are sent to the server module for further SLAM processing. Although offloading part of the SLAM computations to the edge server reduces the computation workload at the client side, it is insufficient to achieve real-time SLAM. On one hand, it may lead to excessive communication latency since huge amount of keyframe-related data needs to be transmitted; On the other hand, the client module is computation-intensive that may not be affordable for resource-constrained mobile platforms. To overcome these limitations, we propose a new architecture shown in Fig. 1 for edge-assisted SLAM that is supported by an adaptive communication strategy, which evaluates the tile importance at the client side and adapts the compression configurations to the importance levels and network condition.

2 SOLUTION

To enable low-latency communications, we design a regional feature prediction module for importance evaluation of sensory data and a configuration adaptation module for adaptive compressing/decompressing. At the client side, the regional feature prediction module identifies the possible feature regions at each future frame so that indicates their importance to SLAM processing. With this importance information, the diverse compressing/decompressing configurations can be generated for those regions so that they can be transmitted in different qualities. As such, the regional feature prediction procedure is pivotal for accurate feature extraction at the edge and bandwidth cost saving.

2.1 System Architecture Overview

2.1.1 Regional Feature Prediction Module. An important characteristic of video frames captured by on-board cameras in mobile

CSCAIoTSys: Coupling of Sensing Computing in AIoT Systems, May 13, 2024, HongKong, China, 15.00
© 2024 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

979-8-3503-6338-8/24/$31.00 ©2024 IEEE
DOI 10.1109/CSCAIoT62585.2024.00011

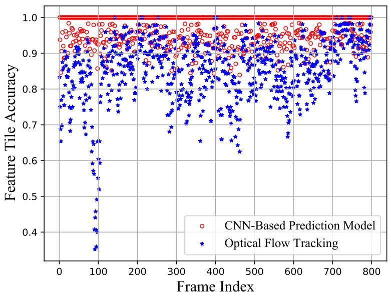

Figure 2: Regional Feature Prediction

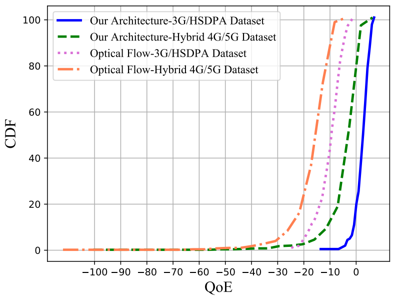

Figure 3: QoE Results

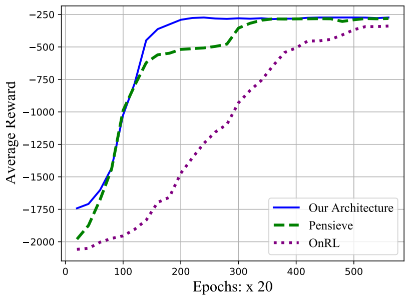

Figure 4: Training Convergence

robotic systems is temporal correlation, based on which, the regional feature prediction module is designed for the importance prediction of the input video frames. Specifically, we partition each video frame into small tiles, and estimate the tile importance level according to the historical knowledge. The principle of tile partition is enlightened from "*detect + track*" framework, which facilitates the detection and tracking of special regions in consecutive frames.

2.1.2 Encoder/Decoder. A pair of encoder/decoder is applied to compress/decompress the tiles of a video frame. The compression configurations are selected by exploiting two types of redundancy in a video frame, namely the intra-tile redundancy and inter-tile redundancy, in order to reduce the frame size. Specifically, the intra-tile redundancy refers to the information that is irrelevant with the SLAM processing in each tile, which is captured by the tile importance level determined by the regional feature prediction module. Based on the intra-tile redundancy, we convert the tiles into various qualities with different resolution. The inter-tile redundancy originates from the intrinsic spatial redundancy in a video frame of natural scenes, which can be mitigated using existing video coding techniques (e.g., H.264) to encode the tiles from a frame into a video segment. In this way, the quantization parameter can be configured according to the tile importance levels to further reduce the communication overhead. For convenience, we denote the compression configurations as a two-tuple, i.e., $< resolution, quantization\ parameter >$. We train a A3C-based Deep Reinforcement Learning (DRL) algorithm to make configuration decisions.

3 RESULTS

3.1 QoE Results

We leverage ORB-SLAM2 as the backbone SLAM and evaluate the Quality-of-Experience (QoE) performance in different network traces. QoE is defined to guide the decision-making of the configuration adaptation algorithm by balancing bandwidth dynamics and the SLAM-related performance. Fig. 3 shows the QoE results obtained by our prediction model and OF tracking from two network traces. Firstly, the performance advantage of our prediction model is further verified according to the comparison with OF tracking. Secondly, Fig. 3 also verifies the effectiveness of the configuration adaptation module in different network environments. From Fig. 3, it can be observed that the optimal QoE results are obtained from the real-measured network traces. This means that in the network environment with low-quality bandwidth, our system still achieve excellent performance by making adaptive compression configurations.

3.2 Convergence Performance

In this experiment, we evaluate the convergence performance of A3C training in the configuration adaptation module. Particularly, we train the DRL agent by using different neural network architectures and then compare the convergence rate, as shown in Fig. 4.

Specifically, the neural network architecture in our configuration adaptation module is shown in Fig. **??**, where the FC layer with 128 elements is taken as the main structure in both policy network and critical network. For performance comparison, we also configure different neural network architectures based on two state-of-the-art applications of the A3C algorithm from Pensieve[3]. Pensieve is a famous adaptive bitrate (ABR) selection system based on A3C, while OnRL is a recent application of the A3C algorithm in live video. Both of them adopt the original learning principle of the A3C algorithm but use difference neural network architectures. The neural network architecture in Pensieve includes a combination of both a 1D-CNN layer a FC layer with 128 elements while OnRL contains only a FC layer with 64 elements and a FC layer with 32 elements. From the results in Fig. 4, it is observed that the neural network architecture with FC layers is more suitable in our system.

4 ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China No. 62302396.

REFERENCES

[1] Hugh Durrant-Whyte and Tim Bailey. 2006. Simultaneous localization and mapping: Part I. *IEEE robotics & automation magazine* 13, 2 (2006), 99–110.

[2] Hyunbum Kim, Jalel Ben-Othman, Lynda Mokdad, Junggab Son, and Chunguo Li. 2020. Research Challenges and Security Threats to AI-Driven 5G Virtual Emotion Applications Using Autonomous Vehicles, Drones, and Smart Devices. *IEEE Netw.* 34, 6 (2020), 288–294.

[3] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 197–210.

[4] Jingao Xu, Hao Cao, Danyang Li, Kehong Huang, Chen Qian, Longfei Shangguan, and Zheng Yang. 2020. Edge assisted mobile semantic visual SLAM. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 1828–1837.