engineering proceedings

MDPI

Proceeding Paper

A Novel Method to Improve the Efficiency and Performance of Cloud-Based Visual Simultaneous Localization and Mapping †

Omar M. Salih [1,2], Hussam Rostum [1] and József Vásárhelyi [1,*]

[1] Institute of Automation and Info-Communication, University of Miskolc, 3515 Miskolc-Egyetemváros, Hungary; omar.salih@uni-miskolc.hu (O.M.S.); ; hussam.rostum@uni-miskolc.hu (H.R.)
[2] Technical College of Kirkuk, Northern Technical University, Kirkuk 36001, Iraq
* Correspondence: jozsef.vasarhelyi@uni-miskolc.hu
† Presented at the Sustainable Mobility and Transportation Symposium 2024, Győr, Hungary, 14–16 October 2024.

Abstract: Since Visual Simultaneous Localization and Mapping (VSLAM) inherently requires intensive computational operations and consumes many hardware resources, these limitations pose challenges to implementing the entire VSLAM architecture within limited processing power and battery capacity. This paper proposes a novel solution to improve the efficiency and performance of exchanging data between the unmanned aerial vehicle (UAV) and the cloud server. First, an adaptive ORB (oriented FAST and rotated BRIEF) method is proposed for precise tracking, mapping, and re-localization. Second, efficient visual data encoding and decoding methods are proposed for exchanging the data between the edge device and the UAV. The results show an improvement in the trajectory RMSE and accurate tracking using the adaptive ORB-SLAM. Furthermore, the proposed visual data encoding and decoding showed an outstanding performance compared with the most used standard JPEG-based system over high quantization ratios.

Keywords: ORB-SLAM; UAV; mapping; localization; tracking; visual data compression; edge-assisted computing

check for updates

Citation: Salih, O.M.; Rostum, H.; Vásárhelyi, J. A Novel Method to Improve the Efficiency and Performance of Cloud-Based Visual Simultaneous Localization and Mapping. *Eng. Proc.* **2024**, *79*, 78. https://doi.org/10.3390/engproc2024079078

Academic Editors: András Lajos Nagy, Boglárka Eisinger Balassa, László Lendvai and Szabolcs Kocsis-Szürke

Published: 11 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Autonomous navigation systems estimate, perceive, and comprehend their surroundings to accomplish tasks such as path tracking, motion planning, obstacle avoidance, and target detection [1,2]. Over recent years, researchers have devised various localization approaches specifically for UAVs, including vision-based techniques. VSLAM can effectively reduce drift in state estimation by revisiting the previously mapped regions since it generates a globally consistent map estimate [3,4]. However, VSLAM encounters complex computations and consumes massive hardware resources due to the extensive image processing algorithms for detecting, matching, tracking, and mapping the features of the captured surrounding environment. To reduce size and save energy, recent studies have researched offloading computationally extensive VSLAM processing tasks to edge cloud platforms, while the substantial tasks are kept on the mobile device. The offloaded functionalities often comprise optimizing maps through local and global bundle adjustment and detecting or removing unnecessary information from maps [5–7]. Many studies in the literature have explored edge-assisted VSLAM techniques, focusing on transferring data from the mobile device to the edge cloud. Such techniques include Cloud framework for Cooperative Tracking and Mapping (C^2TAM) [8], Swarm Map [9], and collaborative VSLAMs [10,11]. The partitioned VSLAM demands special attention to exchanging the visual information between the UAV and edge cloud using efficient encoding and decoding algorithms. These algorithms should provide the highest compression ratios and shorter execution times and perceive visual data quality required for VSLAM operation [12]. The contribution of this paper is as follows:

- A novel adaptive ORB-SLAM workflow is introduced to enrich the number of strongly detected features. Additionally, it maintains accurate tracking and mapping operations of VSLAM.
- Efficient visual data encoding and decoding methods are developed, which can be effectively integrated within the VSLAM architecture to achieve a higher compression ratio without adversely affecting visual data quality.

The paper is organized as follows: Section 2 presents the adaptive ORB-SLAM method. Section 3 explains the proposed visual data encoding and decoding methods. The results are illustrated in Section 4. Finally, conclusions and suggestions are given in Section 5.

2. Adaptive ORB-SLAM Method

This section explains the proposed adaptive VSLAM (for a detailed explanation of the traditional VSLAM, the reader is referred to [13]). To enhance the performance of the tracking and mapping process in the VSLAM system, the first solution proposes updating the ORB algorithm, which is used in the detection and extracting of the features. Figure 1 presents the flowchart of the feature-based V-SLAM step by step, starting from preparing the video and ending with loop closure.

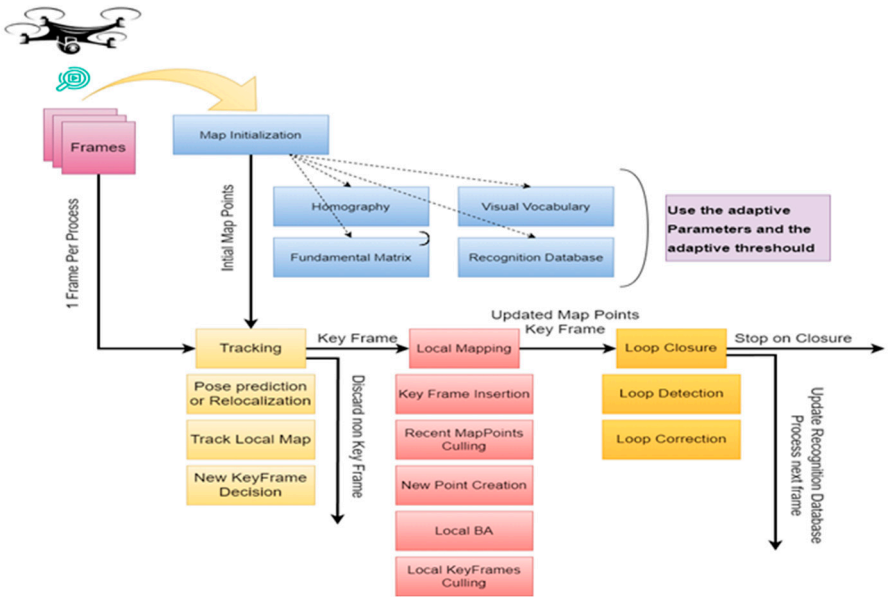


Figure 1. Adaptive ORB-SLAM architecture.

The mounted camera on the UAV captures video footage, which is then processed to extract frames and adjust them, resulting in a resolution of 640 × 480 pixels. The output of this process yields a collection of frames, forming the database for this work. Next, the detect and extract feature will take place. By adaptively selecting the appropriate parameters, the process of detecting and extracting features will repeat until finding suitable parameters that achieve the highest number of the extracted features. After finishing the process of adjusting parameters, the process of matching features will start. This process will continue until the loop closes. The proposed adaptive ORB-SLAM method introduces a new algorithm for VSLAM. These adaptive adjustments increase the number of map points per frame using a strict adaptive threshold to ensure robust tracking and mapping and, as a result, benefit autonomous UAV navigation. Conversely, traditional methods that use fixed parameters (scale factor, the number of levels, and the number of points) are effective in environments containing strong features and minimal change. However, in dynamic environments with frequent changes, adaptive parameters prove to be more effective.

3. Visual Data Encoding and Decoding Methods

The substantial volume of real-time visual data can rapidly exhaust computational resources, particularly in environments characterized by limited resources. Consequently, compressing visual data is essential to mitigating the effects of processing bottlenecks and reducing energy consumption. This compression reduces overall data size, facilitating more efficient storage, bandwidth utilization, and computational processes. Furthermore, compressed visual data accelerate communication between distributed systems, such as edge devices and cloud servers, thereby enabling real-time collaboration and decision-making. Figure 2 illustrates the block diagrams of the proposed encoding and decoding methods. Typically, the encoding system is deployed on the UAV to compress captured data, while the decoding system is deployed on the edge device to reconstruct the original data.

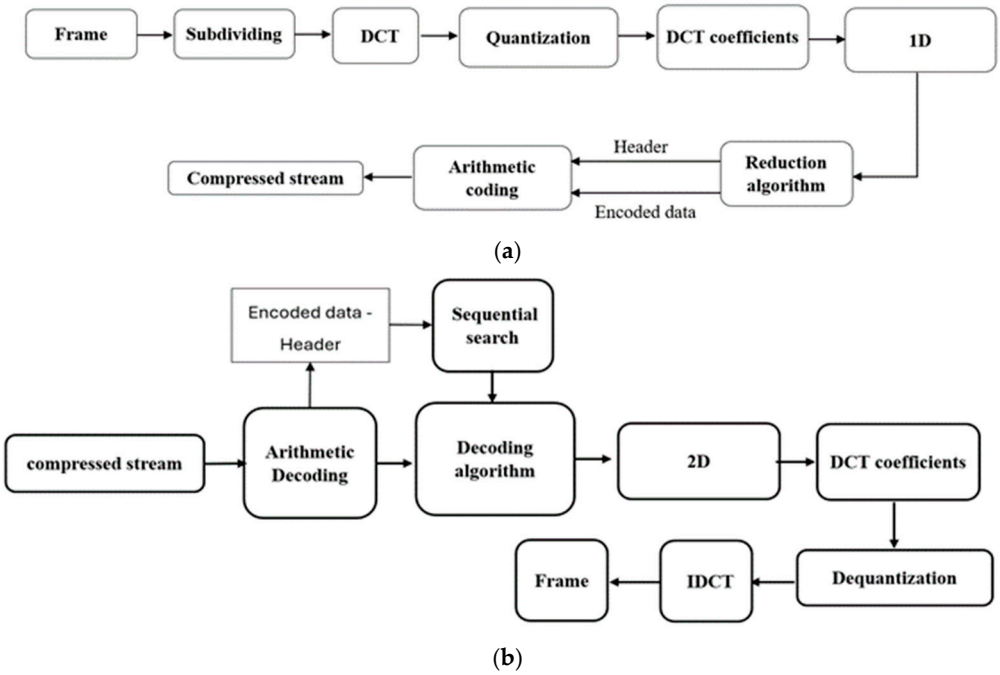


Figure 2. The proposed visual data algorithms for (**a**) encoding and (**b**) decoding.

The proposed methods employ discrete cosine transforms (DCT) as a transformation to frequency domain for its simplicity and effectiveness followed by quantization. The DCT forward and inverse transforms are given in Equations (1) and (2), respectively [14].

$$F(u,v) = \frac{2}{\sqrt{nm}} c(u)c(v) \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2n}\right) \cos\left(\frac{(2y+1)v\pi}{2m}\right), \quad (1)$$

$$f(x,y) = \frac{2}{\sqrt{nm}} \sum_{u=0}^{n-1} \sum_{v=0}^{m-1} c(u)c(v)F(u,v) \cos\left(\frac{(2y+1)v\pi}{2m}\right) \cos\left(\frac{(2x+1)u\pi}{2n}\right), \quad (2)$$

where $0 \le u \le n-1$, $0 \le v \le m-1$, $0 \le x \le n-1$, $0 \le y \le m-1$, and $c(u)$ and $c(v)$ are given below:

$$c(u) = c(v) = \begin{cases} \frac{1}{\sqrt{2}} \; if \; u = 0, v = 0 \\ 1 \; if \; u > 0, v > 0 \end{cases} \quad (3)$$

A proposed coefficient reduction algorithm is implemented to further reduce the quantized coefficients based on three random keys, as shown in the following equation:

$$ED_i = \sum_{j=1}^{3} D_j \cdot K_j, \quad (4)$$

where K_1, K_2, K_3 are the encoding keys, ED_1, ED_2, …., ED_i are the encoded data, D_1, D_2, …, D_j are the input data, and *i,j* are the indices of encoded data and input data, respectively. This algorithm effectively reduces every three coefficients into one single coefficient with the aid of encoding keys. The decoding algorithm works on the edge device using the same steps but in a reversed manner. A fast binary search algorithm is implemented to retrieve the original reduced data with the aid of the same keys used in the encoding phase.

4. Experimental Results

The presented system is implemented and simulated on MATLAB R2022a (MathWorks, Natick, MA, USA), running on an Intel i7-12700H processor (Intel Corporation, Santa Clara, CA, USA) along with 32 GB of RAM. The recorded video frames are resized into 640 × 480 spatial resolution. The following subsections show the results of the enhanced edge-based VSLAM system.

4.1. Testing and Validation of the Adaptive ORB-SLAM Approach

Experimental results show that the use of adaptive ORB parameters and an adaptive threshold enhances the VSLAM algorithm for tracking and mapping. Figure 3 shows the comparison results of the estimated camera trajectory of the proposed method along with the actual camera trajectory; it shows all the map points of the surrounding environment. As a result, by computing the root-mean-square error (RMSE) of trajectory estimates, the adapted threshold and parameters get the absolute RMSE of 0.20123 for the key frame trajectory and 2.6907 for the old algorithm, which depends on fixed parameters.

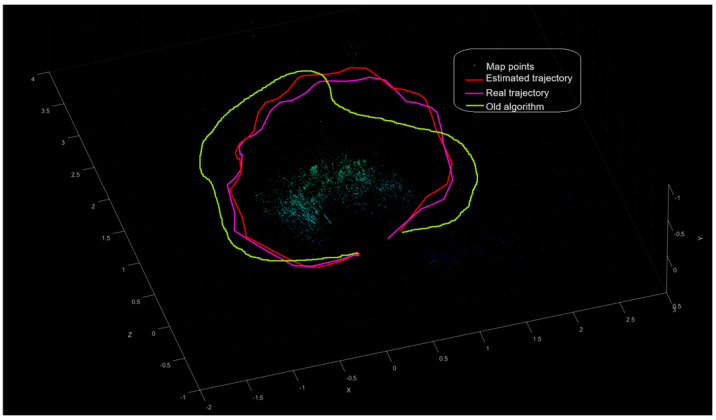


Figure 3. The graph of the comparison between the estimated trajectory and the actual camera trajectory.

The results illustrating the performance of both the adaptive and conventional ORB methods in a dynamic environment are shown in Table 1. A potential downside of using the proposed method with adaptive parameters results in increased computation time, which tends to be longer compared with other methods. Compared with the new method, the computational time is approximately (3 min 15.56 s) to complete the process from start to finish, including tracking and plotting the uploaded ground truth data and comparing the RMSE. In contrast, the traditional ORB feature detection and extraction method takes about (02 min 19.59 s) for the same process.

Table 1. A comparison between the conventional and the proposed adapted algorithm.

Test No.	Scale Factor	Num Levels	Num Points	Threshold	RMSE	Performance Time [min:s]
Traditional algorithm	1.2	8	600	40	2.0697	02:19
Test1	1.4	9	800	50	2.012	03:10
Test2	1.5	10	1000	55	1.9987	04:15
test3	1.6	11	1200	60	1.898	03:50
Adapted method	adaptive	adaptive	adaptive	adaptive	0.20123	03:15

4.2. Visual Data Encoding and Decoding

The proposed algorithms of encoding and decoding are integrated within the VSLAM architecture. The captured frames are processed and encoded on the UAV. The encoded data are sent to the edge for implementing the rest of the VSLAM tasks. Table 2 shows the encoding and decoding performance of the implemented system. The result is compared to a traditional VSLAM architecture employing the standard JPEG, which is the most used in machine vision applications due to its simplicity [15]. The results demonstrate outstanding performance compared with identical VSLAM employing JPEG compression. The proposed system showed a robust operation against server quantization values (Q), and it still worked properly up to Q = 100. After this value (i.e., Q = 125), the proposed system failed to find the trajectory correctly. In contrast, the VSLAM system employing the JPEG failed to estimate the trajectory up to Q = 25. This is attributed to the additive JPEG compression noise, which can be seen clearly through the affected values of the structured similarity index measure (SSIM) and the peak-to-noise ratio (PSNR). However, the execution time of the proposed method is still higher than for the corresponding JPEG-based system due to the additional blocks for reducing the data. Nevertheless, the execution time gets closer at a higher compression ratio, which could significantly decrease this difference. Harnessing the high capabilities of cloud platforms could speed up the decoding execution time. Moreover, an efficient hardware implementation could also decrease the execution time of the encoding system using advanced techniques such as parallel processing techniques rather than sequential operation in MATLAB.

Table 2. A sample frame performance comparison between the proposed and JPEG-based VSLAM.

Sample Frame No.	Original Size [KB]	Q	VSLAM-Based Proposed Method					
			Compressed Size [KB]	PSNR	SSIM	Compression Ratio %	Encoding Time [s]	Decoding Time [s]
1029	439	10	14.679	34.2990	0.9420	96.7	1.0373	39.9312
		15	12.344	33.3807	0.9343	97.2	0.8941	11.7414
		20	10.918	32.8027	0.9286	97.5	0.8686	5.4861
		25	9.963	32.4911	0.9234	97.7	0.8017	3.1809
		50	7.277	31.6654	0.9013	98.3	0.7006	1.1124
		75	5.881	30.7682	0.8789	98.7	0.7376	0.9004
		100	4.997	29.6547	0.8596	98.9	0.6805	0.8666
		125	-	-	-	-	-	-
		VSLAM-Based Standard JPEG						
		Q	**Compressed Size [KB]**	**PSNR**	**SSIM**	**Compression Ratio %**	**Encoding Time [s]**	**Decoding Time [s]**
		10	39.4	29.7012	0.8403	91.0	0.6012	0.6071
		15	31.6	26.9786	0.8031	92.8	0.3417	0.5327
		20	25.8	24.8070	0.7685	94.1	0.3134	0.5309
		25–125	-	-	-	-	-	-

5. Conclusions

In this paper, a novel method for enhancing the localization and mapping process in the VSLAM algorithm in dynamic environments is presented. The enhancement was achieved using adaptive ORB parameters instead of fixed ones during the map initialization step. These parameters automatically adjust to achieve optimal detection and feature extraction. The results from this step are used in feature matching between frames, which is also optimized through an adaptive threshold. These improvements followed by proposed encoding and decoding algorithms enable the UAVs to deploy in low-scale resource devices. The experimental results show a percentage decrease in the RMSE of around 92.52% for key frame trajectory (in meters) compared with the traditional algorithm. Additionally, it shows that the proposed method remains robust up to 98.9% of input frame compression ratio with better PSNR and SSIM compared with the conventional JPEG-based architecture. This research can be extended to optimize the proposed methods for seamless multi-device synchronization, enabling smooth transitions between virtual and physical worlds in metaverse applications. Additionally, developing lightweight algorithms and efficient hardware implementations ensures high-quality immersive experiences on resource-constrained devices like AR (Augmented Reality) and VR (Virtual Reality) headsets, without compromising performance or battery life. Finally, implying the proposed methods in edge–cloud collaboration will help support large-scale virtual environments with low latency.

Author Contributions: Conceptualization, O.M.S., H.R. and J.V.; methodology, O.M.S., H.R. and J.V.; validation, O.M.S., H.R. and J.V.; formal analysis, J.V., O.M.S. and H.R.; investigation, O.M.S., H.R. and J.V; resources, O.M.S., H.R., and J.V; writing—original draft preparation O.M.S., H.R. and J.V.; writing—review and editing O.M.S., H.R. and J.V.; visualization, O.M.S., H.R. and J.V.; supervision J.V.; project administration J.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rostum, H.M.; Vásárhelyi, J. A review of using visual odometery methods in autonomous UAV Navigation in GPS-Denied Environment. *Acta Univ. Sapientiae Electr. Mech. Eng.* **2023**, *15*, 14–32. [CrossRef]
2. Al-Tawil, B.; Hempel, T.; Abdelrahman, A.; Al-Hamadi, A. A review of visual SLAM for robotics: Evolution, properties, and future applications. *Front. Robot. AI* **2024**, *11*, 1347985. [CrossRef] [PubMed]
3. Chen, W.; Shang, G.; Ji, A.; Zhou, C.; Wang, X.; Xu, C.; Li, Z.; Hu, K. An overview on visual slam: From tradition to semantic. *Remote Sens.* **2022**, *14*, 3010. [CrossRef]
4. Bouhamatou, Z.; Abdessemed, F. Visual Simultaneous Localisation and Mapping Methodologies. *Acta Mech. Autom.* **2024**, *18*, 451–473. [CrossRef]
5. Cui, X.; Lu, C.; Wang, J. 3D semantic map construction using improved ORB-SLAM2 for mobile robot in edge computing environment. *IEEE Access* **2020**, *8*, 67179–67191. [CrossRef]
6. Chase, T.; Ben Ali, A.J.; Ko, S.Y.; Dantu, K. PRE-SLAM: Persistence Reasoning in Edge-assisted Visual SLAM. In Proceedings of the 2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 18–21 October 2022; pp. 458–466.
7. Dechouniotis, D.; Spatharakis, D.; Papavassiliou, S. Edge robotics experimentation over next generation iiot testbeds. In Proceedings of the 2022 IEEE/IFIP Network Operations and Management Symposium (NOMS), Budapest, Hungary, 25–29 April 2022; pp. 1–3.
8. Riazuelo, L.; Civera, J.; Martinez Montiel, J.M. C2tam: A cloud framework for cooperative tracking and mapping. *Robot. Auton. Syst.* **2014**, *62*, 401–413. [CrossRef]
9. Xu, J.; Cao, H.; Yang, Z.; Shangguan, L.; Zhang, J.; He, X.; Liu, Y. SwarmMap: Scaling up real-time collaborative visual SLAM at the edge. In Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), Renton, WA, USA, 4–6 April 2022; pp. 977–993.

10. Mohanarajah, G.; Usenko, V.; Singh, M.; D'Andrea, R.; Waibel, M. Cloud-based collaborative 3D mapping in real-time with low-cost robots. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 423–431. [CrossRef]
11. Eger, S.; Pries, R.; Steinbach, E. Evaluation of different task distributions for edge cloud-based collaborative visual SLAM. In Proceedings of the 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), Tampere, Finland, 21–24 September 2020; pp. 1–6.
12. Salih, O.M.; Vásárhelyi, J. Visual Data Compression Approaches for Edge-Based ORB-VSLAM Systems. In Proceedings of the 2024 25th International Carpathian Control Conference (ICCC), Budapest, Hungary, 27–30 May 2024.
13. Mur-Artal, R.; Martinez Montiel, J.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
14. Mukherjee, D. Parallel implementation of discrete cosine transform and its inverse for image compression applications. *J. Supercomput.* **2024**, *in press*. [CrossRef]
15. Hamano, G.; Imaizumi, S.; Kiya, H. Effects of jpeg compression on vision transformer image classification for encryption-then-compression images. *Sensors* **2023**, *23*, 3400. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.