Jurnal Polimesin
Department of Mechanical Engineering
State Polytechnic of Lhokseumawe
http://e-jurnal.pnl.ac.id/polimesin

e-ISSN : 2549-1999 No. : 2 Month : April
p-ISSN : 1693-5462 Volume : 22 Year : 2024

Article Processing Dates: Received on 2024-02-21, Reviewed on 2024-02-28, Revised on 2024-04-23, Accepted on 2024-04-26 and Available online on 2024-04-30

Visual place recognition for autonomous mobile robot navigation using LoFTR and MAGSAC++

Udink Aulia[1,2], Iskandar Hasanuddin[2], Muhammad Dirhamsyah[2], and Nasaruddin[3*]

[1]Doctoral Program, School of Engineering, Post Graduate Program, Universitas Syiah Kuala, Banda Aceh, 23111, Indonesia
[2]Dept. of Mechanical and Industrial Engineering, Universitas Syiah Kuala, Banda Aceh, 23111, Indonesia
[3]Dept. of Electrical and Computer Engineering, Universitas Syiah Kuala, Banda Aceh, 23111, Indonesia
*Corresponding author: nasaruddin@usk.ac.id

Abstract

Autonomous mobile robots are defined as robotic entities capable of independent movement and intelligent decision-making, relying on their ability to perceive and analyze their surroundings, including objects in their environment. In Simultaneous Localization and Mapping (SLAM) systems, loop closure is often achieved through visual place recognition techniques, where the system compares the current visual input with previously observed scenes to identify matches. In computer vision applications, Speeded-Up Robust Features (SURF) and Scale-Invariant Feature Transform (SIFT) are popular feature extraction algorithms used for such as key point detection, matching, and image registration tasks. The choice of inlier threshold should be based on the specific characteristics of the application and the nature of the images being processed. It often requires experimentation and tuning to find the optimal balance between robustness and accuracy. It Utilizes the pre-trained Local Feature Transformer (LoFTR) and MAGSAC++ estimator to address these drawbacks by employing the number of inliers to determine the similarity between two images for visual place recognition. Our experiment demonstrates that the number of inliers can determine the similarity of locations between two images. Scale variations and translation in location significantly influence the resulting number of inliers. Comparing images from the same location and from different locations yields varying numbers of inliers. The number of inliers significantly influences the similarity of locations. At the same location, the number of inliers is above 150, while at different locations, the number is below 150.

Keywords:
SLAM, LoFTR, inlier, key point, visual place recognition.

1 Introduction

In recent times, there has been substantial interest in Visual Place Recognition (VPR), which involves identifying the location of images. This matter has garnered significant attention across various research domains, including computer vision, robotics, and machine learning [1]. Identifying locations visually is considered a crucial element in localization and navigation, playing a role in loop closure within Simultaneous Localization and Mapping (SLAM) algorithms [2].

Mobile robot navigation systems are often based on SLAM techniques. The crucial duty of finding previously visited locations in SLAM implementations is called "loop closure," in which the system finds a previously visited location and applies this knowledge to correct the map, which is then used to fix map inaccuracies that build up over time [3]. To execute visual SLAM mapping, cameras are used to gather data about the surroundings. Computer vision and odometry techniques are then combined to map the environment [4].

In particular, loop closure is usually solved via VPR in systems where cameras are the primary sensors [5]. It might be difficult to visually identify a place, especially in uncontrolled outside conditions and while doing so for extended periods. This is because images shot at various times in the same location might have quite distinct visual characteristics. The primary factors contributing to the observed variations could stem from alterations in viewpoint and illumination, the cyclic transitions between day and night, fluctuations in seasonal conditions, or the existence of dynamic elements and obstructive elements [6]. Recent advances in computer vision have been made possible by neural networks' capacity to learn feature representations from data that are superior to previously created ones by hand [7], while the most widely used feature descriptor for these kinds of jobs has likely been handmade strong features like SIFT, SURF, ORB, etc. [8]. Because of their accuracy and adaptability, Deep Neural Networks (DNNs) are widely employed in the study of automatic categorization problems. Modern designs and pre-trained networks are already available and may be modified and improved to tackle more recent classification jobs. These networks often consist of many layers that are coupled to one another and have numerous parameters per layer. One family of deep neural networks called Convolutional Neural Networks (CNN) is most frequently used to analyze visual images. CNN has shown remarkable effectiveness in several computer vision tasks, which has led to the most recent trend in VPR research [9]. As a result, several authors have used the activations of specific CNN layers to provide visual representations that are appropriate for addressing the VPR issue [10]. Visual place recognition can be challenging due to various issues, such as perceptual aliasing. Places are not always revisited from the same viewpoint and position as before, and environments where multiple places may appear strikingly similar, with appearances that can change dramatically [11].

In particular, visual location identification plays a crucial role in visual Simultaneous Localization and Mapping (vSLAM) loop closure detection, which removes accumulated errors. Furthermore, precise pose and map in vSLAM systems require a strong tracking module. However, in real-world applications, tracking failure is unavoidable because of factors including quick motion, hazy images, unexpected shifts in the camera's visual perspective, texture deficiency, etc. As a result, a powerful relocalization module is essential.

Two key elements are used in contemporary feature-based vSLAM systems to relocalize the robot. Visual place recognition (finding potential keyframes) is the first phase, and key point feature matching (metric localization) is the second phase. In large-scale localization and mapping, trajectory drift and the construction of an unclear map of an unknown environment are inevitable without precise visual place recognition [12]. The foundation of many 3D computer vision applications, such as SLAM, is local feature matching between images. The majority of matching techniques now in use split an image into three stages: feature matching, feature description, and feature detection [13].

During the detection stage, each image's important features, such as corners, are initially identified as interest points. After that, local descriptors are retrieved from the areas surrounding these interest sites. Two sets of interest points with descriptors are produced during the feature detection and description phases. The point-to-point correspondences between these sets of points are then discovered using more complex matching algorithms like closest neighbor search [14]. Using a feature detector narrows the matching search space, and the resulting sparse correspondences are enough for the majority of applications, such as estimating camera posture. However, due to a variety of issues, including inadequate texture, repeating patterns, perspective changes, lighting variations, and motion blur, a feature detector may not be able to extract enough recurrent interest spots across images [15].

In mobile robotics, autonomous navigation is crucial, as it allows robots to navigate and adapt to complex and dynamic environments. A visual map scheme is a map of images that serves as the basis for a robot navigation strategy that employs a vision system as its only sensor. This strategy's representation can be thought of as a topological map, with key images serving as the visual cues for each node in the environment [16]. The learning phase will be the main emphasis of this study to extract the best possible visual map from the surroundings for visual localization and self-navigating.

Successful vSLAM implementations assess the effectiveness and efficiency of popular feature detectors and descriptors like SIFT, SURF, ORB, BRISK, and AKAZE in matching consecutive images, alongside algorithms like Nearest Neighbor (NN) for keypoint set matching and Homography based on RANSAC to reject outliers, particularly in underwater environments [17].

The study analyzes four feature detection and description methods—ORB, BRISK, KAZE, and Accelerated KAZE—and three outlier rejection methods—RANSAC, GC-RANSAC, and MAGSAC++. The analysis involves both visual and quantitative assessments to determine the most accurate and robust registration method for the histopathological dataset. Evaluation metrics such as the number of detected key points and inliers are used to evaluate the performance of different pairs of feature detection-description methods and outlier rejection algorithms [18].

In the realm of state-of-the-art advancements, a hardware implementation of the ORB algorithm on a heterogeneous SoC FPGA device. To validate hardware design, the researcher performed a comparative analysis against a software model. This software model was developed using functions from the OpenCV library, including feature point matching from the FLANN submodule, the RANSAC algorithm for inlier identification, and the computation of the homography matrix. Both the outputs from OpenCV's ORB and hardware implementation were used as inputs to the software model, enabling a comprehensive evaluation based on key metrics such as the number of inliers, matching rate, rotation error, and translation error [19].

As far as the authors are aware, there is limited documentation that thoroughly examines the use of feature detectors and descriptors for mobile robot navigation, specifically employing Local Feature Transformer (LoFTR) and the MAGSAC++ estimator to match two images based on the number of inliers.

A novel method is proposed based on a detector-free approach to local feature matching that enables the production of feature-rich visual maps for outdoor situations. To generate a visual control policy between two consecutive key images, our deep learning approach uses pre-trained Local Feature Transformer (LoFTR) [20]. Additionally, a geometry constraint is used to ensure that the images share features using the MAGSAC++ [21] estimator to match two images with different perspectives of the shot and identify the same key points. Moreover, in this paper, this paper investigates the impact of the number of inliers on the similarity of key images under shift, scale, and occlusion circumstances, conducting extensive tests and making quantitative and visual comparisons between images. The next step is to calculate the least number of inliers for images when the locations are about the same and the maximum number of inliers to show that the two images are not the same.

2 Research Methods

LoFTR, short for Local Feature-based Transformer, is a recent approach that combines local feature matching with transformer networks for accurate and efficient visual localization tasks. It leverages the strengths of both traditional feature-matching techniques and modern deep learning approaches to achieve state-of-the-art performance in tasks such as visual odometry and Simultaneous Localization and Mapping (SLAM). MAGSAC++, on the other hand, stands for marginalizing sample consensus. It is an improved version of the RANSAC algorithm, which is commonly used for outlier rejection in computer vision tasks. MAGSAC++ provides a more robust estimation of model parameters by marginalizing the outlier probabilities, leading to more accurate and reliable results, especially in challenging environments with a high percentage of outliers. Together, LoFTR and MAGSAC++ form a powerful combination for matching images in scenarios like mobile robot navigation, where accurate feature matching and outlier rejection are crucial for reliable performance.

This research method has 4 stages, including the selection of images from the dataset, the selection of detectors for local feature matching, the selection of estimators, and visual place evaluation.

2.1 Dataset

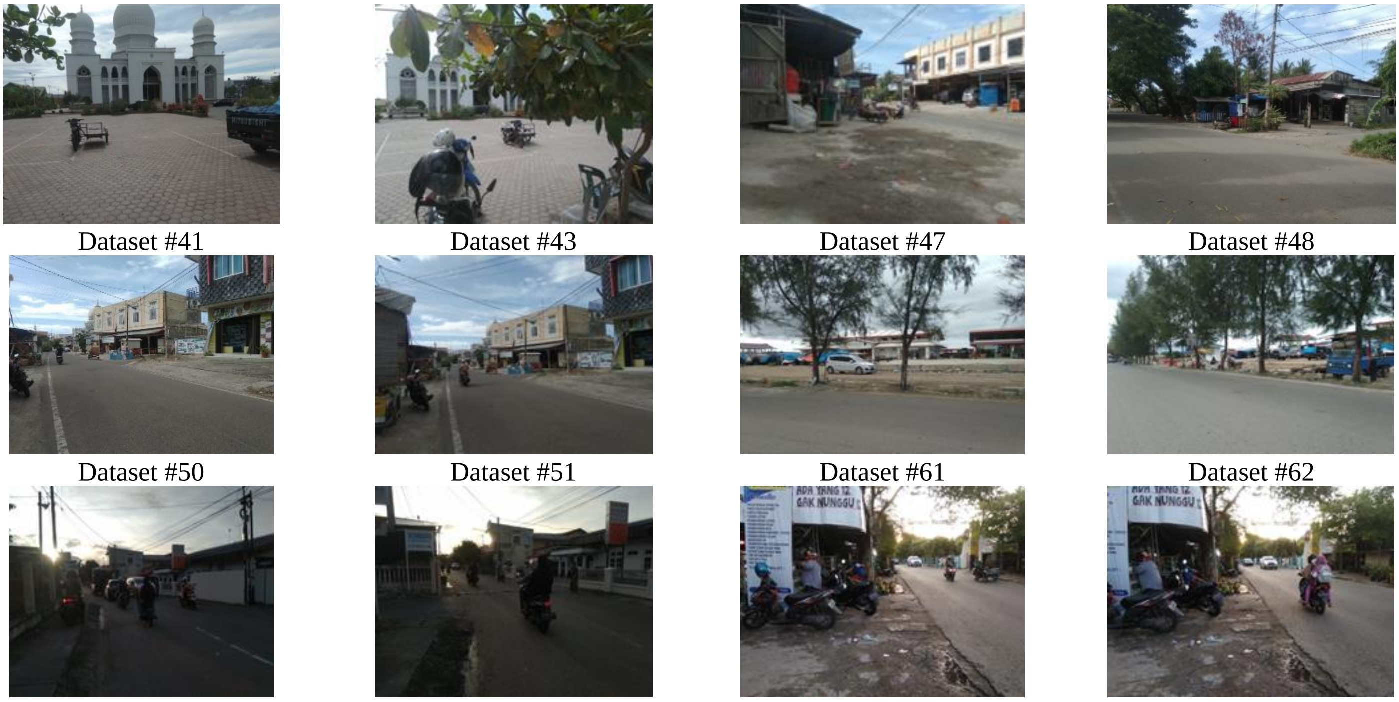
The dataset used is the Banda Aceh city dataset which has 400 images with resolution 800×600 pixels. The Banda Aceh city dataset features daytime traverses captured in an urban setting by a motorcyclist, resulting in modest condition differences and considerable perspective alterations. Many dynamic elements, including moving and parked cars, pedestrians, goods rickshaws, and motorcycles, are present in this metropolitan setting. The majority of the time, there is also an abundance of greenery that obscures more static and distinctive structures like buildings.

Sixteen images selected from this dataset, as depicted in Fig. 1, are used to evaluate visual place recognition between two images by considering the number of inliers. The number of inliers will determine the similarity of places based on the scale and translation of the two images.

2.2 Pre-Trained Detector-free Local Feature Matching

End-to-end local feature matching is accomplished using the detector-free deep neural network architecture known as a pre-trained Local Feature Transformer (LoFTR) [20]. The LoFTR indoor model has been trained using the ScanNet [22] dataset and the outdoor model on the MegaDepth [23]. ScanNet has 230 million image pairings for training and 1613 monocular sequences with ground truth postures and depth maps. One million online images from 196 distinct outdoor scenes make up MegaDepth. Matching amid drastic perspective shifts and repeating patterns is MegaDepth's main challenge. On several datasets, LoFTR produces state-of-the-art results in relative posture estimation and visual localization [20].

Patch embedding, multi-scale transformer network, local feature descriptor, and geometric verification are its four primary parts. First, a multi-level feature vector representation of an image patch is computed by the patch embedding component. Second, to obtain local feature information at various scales, the multi-scale transformer network processes the feature vector of picture patches. Subsequently, the transformer network output is mapped to a fixed-length feature descriptor by the local feature descriptor, which records the i-th image patch's local geometry and appearance. After extracting feature descriptors from both images, geometric verification finally calculates a homography matrix that aligns two images based on the matched features. On an NVIDIA T4 with 12.67 GB of GPU RAM, the pre-trained model with 11.56 million parameters processes a 640×480 image pair in 116 ms.



Dataset #41 Dataset #43 Dataset #47 Dataset #48

Dataset #50 Dataset #51 Dataset #61 Dataset #62

Dataset #179 Dataset #180 Dataset #200 Dataset #201

Fig 1. Selected dataset for evaluation.

In this research, to achieve location similarity between two images, we utilize a division of inlier regions into low, medium, and high categories. The high region represents areas with nearly identical locations, no obstacles, and equal distances from the camera with low differences in translation and scale. The number of inliers for the high region is greater than 1000. The medium region comprises areas where locations are similar, but some parts of the image are obstructed or have different scales, resulting in not all key points in image 1 matching with image 2. The number of inliers for the medium region is greater than 150 and less than 1000. The low region includes areas where the locations of image 1 and image 2 are dissimilar, with the lowest number of matching inliers less than 150. The use of a range for confidence values is due to pre-trained LoFTR and MAGSAC++ having a high number of inliers for similarity between the same two images, allowing for the division into three distinct regions to determine the location similarity between two images.

2.3 Estimator

RANSAC is a popular estimator for several multimedia applications, such as feature matching, and is considered a reliable tool for model fitting [24]. In addition to its ease of implementation, RANSAC has the appealing feature of having few adjustment parameters. However, when dealing with expansive baseline scenarios, multiple viewpoints, flexible movements, and a significant number of discrepancies in the assumed correspondences, RANSAC also faces efficiency and robustness issues. To remove the threshold from the model quality computation, Barath et al. [25] presented the Marginalizing Sample Consensus technique (MAGSAC), which involves marginalizing over the noise σ.

In addition to not requiring a threshold to be manually established, the MAGSAC method was shown to be much more accurate than existing robust estimators on a variety of issues across many datasets. Modern robust estimators are inferior to MAGSAC++ and P-NAPSAC [26] samplers in terms of rate of failure, velocity, and precision.

2.4 Visual Place Evaluation

Following the completion of the data set image selection, we use pre-trained LoFTR's deep learning technique to compare the images. The tests were carried out in Google Colaboratory and implemented in Python using Kornia [27], a PyTorch Open Source Computer Vision Library. The basic matrix was used in geometric validation together with MAGSAC++ for feature-matching refinement (outlier identification).

The number of images used in the experiment was 16 images with image composition related to scale enlargement and location shift. Images from different locations are compared. In scale enlargement mode, the effect on the number of inliers produced and the influence of the scale enlargement multiplier factor on the number of inliers are examined. For the location shift mode, the effect of the shift distance on the number of inliers produced is investigated. The effect of images being at different locations will also be reviewed to see the maximum number of inliers to state that the two images are not at the same location. In Fig. 1, the images used in the experiment are shown with the image name in the form of a number.

3 Results and Discussion

3.1 Proposed Method

Pre-trained LoFTR is used for local feature matching between images, and MAGSAC++ is employed as a robust estimator to determine the number of inliers in establishing whether the two images share the same location. Sixteen images from this dataset are selected to evaluate visual place recognition between two images by examining the number of inliers. The number of inliers will determine the similarity of places based on the scale and translation of the two images. The experiment will determine areas of inlier values for the same and unequal locations.

The matches show the qualitative outcomes. Lines joining the matching feature points in two images indicate matches. It creates lines that represent the best matches between the first and second images by stacking them horizontally. Quantitative results are displayed in a table that has: the identity of the source image and the destination image, the type of relationship applied to the destination image to the source image, the number of inliers obtained, and the grouping of image similarities in the high, medium and low areas. The grouping results for high and medium are a reference that both images are in the same location while for low they are in different locations.

3.2 Experiment

An experiment was conducted aimed at generating both quantitative and qualitative data. Quantitative data were obtained by assessing the number of inliers during the feature-matching process between image pairs. On the other hand, qualitative data included the color of the matching feature lines. Sixteen images were selected to assess the performance of pre-trained LoFTR in the matching process between two images, which will generate the number of inliers based on the level of similarity—specifically, high, medium, and low. Numbers of inliers were obtained by matching between 2 images using lines connecting key-points. The matches that meet the fundamental matrix constraint are shown in red lines for high confidence value and blue lines for low confidence value otherwise between red and blue lines. Images from the same location are compared, as shown in Fig. 2, displaying 3225 matched points between key images for dataset #200 and dataset #201 obtained with LoFTR+MAGSAC. The red line indicates that many lines have a high confidence value. Pre-trained LoFTR produces a high number of inliers because dataset #200 is shifted 1 meter to the right compared to dataset #201. Then there is no change in scale and viewpoint rotation.

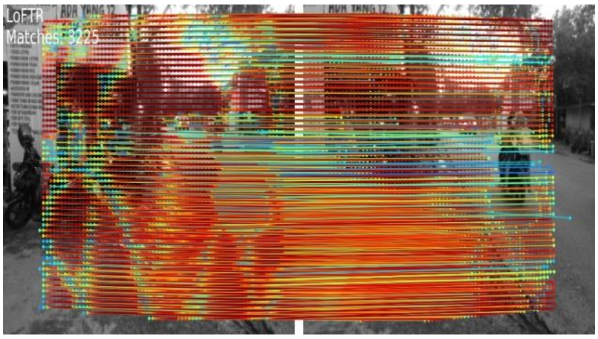


Fig. 2. Matched points between key images for dataset #200 and dataset #201.

The effect of occlusion that occurs in dataset #43 is shown in Fig. 3, displaying 490 matched points between key images for dataset #41 and #43 obtained with LoFTR+MAGSAC.

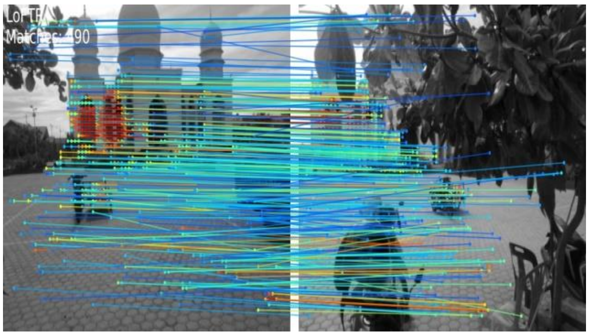


Fig. 3. Matched points between key images for dataset #41 and dataset #43.

The number of inliers produced fell below 1000. There are still red lines connecting between images #41 and #43, indicating that both images are in the same location even though the mosque building is partially closed.

The number of inliers typically refers to the number of matched feature points between two images. During the comparison process, datasets #200 and #201 have the same distance from the camera. They feature a small vehicle obstacle, leading to only a few key points not matching and resulting in 3225 inliers. Datasets #41 and #43 have an obstacle relationship, whereas dataset #43 has a large obstacle in the form of a tree, causing not all key points to match with dataset #41, resulting in 490 inliers. In datasets #179 and #180, there are different distances from the camera at the same location, leading to a scaling process that causes not all key points to match between the two images. When the number of inliers is lower for images with obstacles compared to images without obstacles, it suggests that matching key points between the two images is more challenging in the presence of obstacles. Obstacles can obscure or alter the appearance of visual features in an image. This ambiguity can make it challenging for algorithms to accurately match key points.

Changes in scale or enlargement of objects produce inlier values below 1000 because not all parts of dataset #83 are contained in image #84. Fig. 4 depicts 424 matched points between key images for dataset #83 and dataset #84 obtained with LoFTR+MAGSAC. Several red lines have a high confidence value between dataset #83 and #84, dataset #84 is the result of enlarging the scale of dataset #83 with the center position on the goods rickshaw.

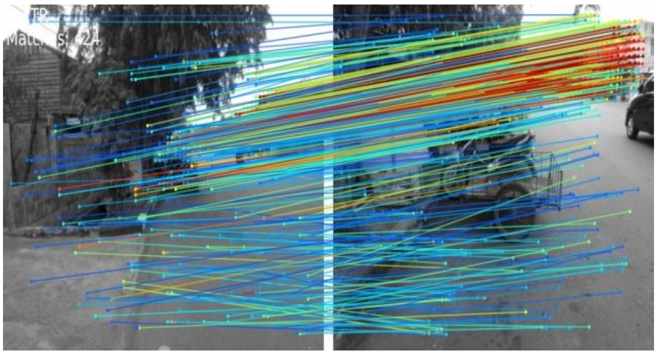


Fig. 4. Matched points between key images for dataset #83 and dataset #84.

Experiment with shifting the image a fairly long distance, as in dataset #62 versus dataset #61. Fig. 5 shows 195 matched points between key images for datasets #61 and #62 obtained with LoFTR+MAGSAC. The left part of dataset #61 is in the middle of dataset #62 and the red line is still visible connecting these two datasets which are still in the same location. This inlier value is the lowest value obtained from image comparisons for the same location.

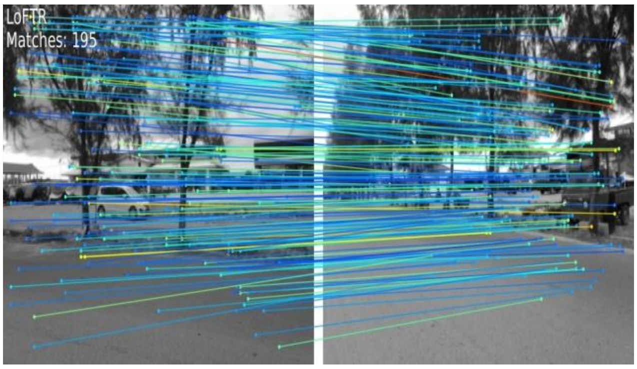


Fig. 5. Matched points between key images for dataset 61 and dataset 62.

The number of inliers produced was also tested when using 2 images with different locations. Fig. 6 displays 133 matched points between key images for datasets #41 and #200 obtained with LoFTR+MAGSAC. The two images are not in the same location. This can be identified by the absence of red lines and the many lines that intersect each other. The number of inliers for dissimilar images is 133.

The effect of location dissimilarity on the number of inliers was tested for different locations. Fig. 7 displays 121 matched points between key images for datasets #47 and #48, which are in different locations. Even though there is a red line, this value is the lowest compared to the number of inliers that have been carried out in previous evaluations. The red line which has a high confidence value still appears even though the two places are not the same.

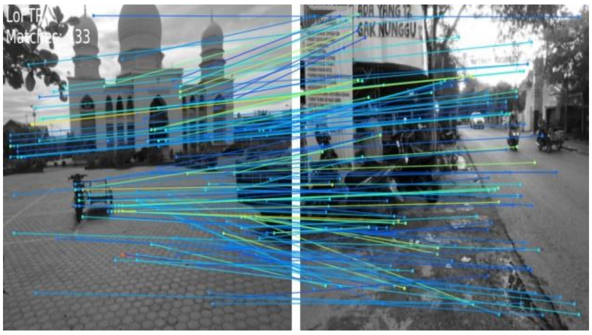


Fig. 6. Matched points between key images for datasets #41 and #200.

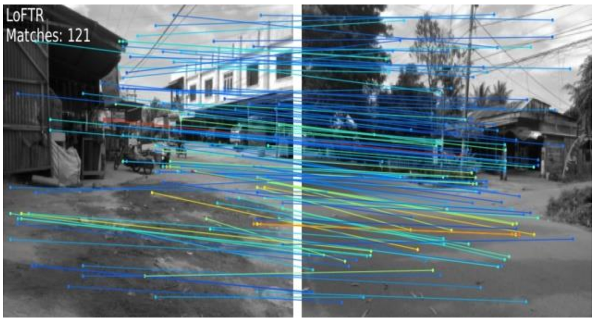


Fig. 7. Matched points between key images for dataset 47 and dataset 48.

Quantitative evaluation results of pre-trained LoFTR are provided in Table 1. In Table 1, there are 3 relationships between the two images, namely trans (translation), obs (obstacle), and scale. Our experiment demonstrates that the number of inliers can determine the similarity of locations between two images. Scale variations and translation in location significantly influence the resulting number of inliers. A count of inliers exceeding 1000 indicates that both images are in the same location with no obstacles and small translations. For images subjected to scale enlargement, viewpoint translation, and obstruction, the number of inliers ranges from 180 to 600. Meanwhile, for images not in the same location, the number of inliers is less than 150. Qualitatively, the appearance of the red line is not related to the similarity or dissimilarity of the locations. Quantitatively, the number of inliers has a big influence on location similarity.

Table 1. Visual place recognition with pre-trained LoFTR for selected Banda Aceh city dataset.

Image1	Image2	Relation	Inliers	Similarity
50	51	trans	3019	high
128	129	trans	2749	high
200	201	obs	3225	high
179	180	scale	580	medium
41	43	scale	490	medium
83	84	scale	424	medium
61	62	trans	195	medium
41	200		133	low
47	48		121	low

An innovative approach to feature matching in local images is introduced. It is suggested that pixel-wise dense matches be established at a coarse level and then refined at a fine level, rather than conducting image feature recognition, description, and matching sequentially. Unlike dense approaches that search correspondences using cost volume, Feature descriptors are extracted conditioned on both images using Transformers' self and cross attention layers. Our technique may generate dense matches in low-texture environments, where feature detectors often fail to create repeating interest sites, thanks to Transformers' global receptive field.

4 Conclusion

This paper introduced a unique detector-free matching technique called LoFTR, capable of constructing accurate semi-dense matches using transformers in a progressive and detailed manner, providing a revolutionary framework for visual location identification for autonomous mobile robots. For LoFTR to achieve high-quality matches on indistinctive regions with poor texture or repeating patterns, the pre-trained LoFTR module uses transformers' self and cross-attention layers to change the local characteristics such that they rely on context and location. The MAGSAC++ estimator and P-NAPSAC sampler are used to generate the number of inliers. Visual place recognition between two images is evaluated by examining the number of inliers, using a selection of 16 images from the Banda Aceh city dataset. The number of inliers will determine the similarity of places based on the scale and translation of the two images. The number of inliers significantly influences the similarity of locations. At the same location, the number of inliers is above 150, while at different locations, the number of inliers is below 150.

Different scales, translations, and obstacles have a great impact on the number of inliers. These effects can result in fewer key points being matched correctly, leading to a decrease in the number of inliers. The image scale represents the distance of the image from the camera; a greater difference in distance between the two images and the camera results in a smaller number of inliers. This is because not all key points match. The influence of translation causes a small number of key points on the sides of the image to not match, yet the number of inliers remains above 1000. In obstacle mode, the size of the obstacle area within the image will determine the number of inliers produced.

Including a quality metric to identify similar characteristics in local images, using the number of inliers, is suggested to address the problem of visual place recognition. Near-perfect identification is achieved across most image pair comparisons using this strategy. Pre-trained LoFTR exhibits improved matching accuracy and decreased mismatches, which effectively tackles issues related to low-texture areas large perspective, and small translation.

Acknowledgments

This work was supported by the Ministry of Education, Culture, Research, and Technology of The Republic of Indonesia Grant No. 575/UN11.2.1/PT.01.03/DPRM/2023.

References

[1] C. Masone and B. Caputo, “A Survey on Deep Visual Place Recognition,” *IEEE Access*, vol. 9, pp. 19516–19547, 2021, doi: 10.1109/ACCESS.2021.3054937.

[2] D. Zhou, Y. Luo, Q. Zhang, Y. Xu, D. Chen, and X. Zhang, “A Lightweight Neural Network for Loop Closure Detection in Indoor Visual SLAM,” *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, pp. 1–11, 2023, doi: 10.1007/s44196-023-00223-8.

[3] C. Theodorou, V. Velisavljevic, V. Dyo, and F. Nonyelu, “Visual SLAM algorithms and their application for AR, mapping, localization and wayfinding,” *Array*, vol. 15, no. July, p. 100222, 2022, doi: 10.1016/j.array.2022.100222.

[4] L. Chen, S. Jin, and Z. Xia, “Towards a Robust Visual Place Recognition in Large-Scale vSLAM Scenarios Based on a Deep Distance Learning,” *Sensors*, vol. 21, 2021, doi: 10.3390/s21010310.

[5] S. Lowry *et al.*, “Visual Place Recognition: A Survey,” *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, 2016, doi: 10.1109/TRO.2015.2496823.

[6] L. G. Camara and L. Přeučil, “Visual Place Recognition by spatial matching of high-level CNN features,” *Rob. Auton.*

Syst., vol. 133, p. 103625, 2020, doi: 10.1016/j.robot.2020.103625.

[7] S. K. Sharma, A. Jain, and A. K. Shukla, "A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching," *Appl. Sci.*, vol. 13, no. 10, 2023, doi: 10.3390/app13106015.

[8] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," *2018 Int. Conf. Comput. Math. Eng. Technol. Inven. Innov. Integr. Socioecon. Dev. iCoMET 2018 - Proc.*, vol. 2018-Janua, pp. 1–10, 2018, doi: 10.1109/ICOMET.2018.8346440.

[9] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier, "A Holistic Visual Place Recognition Approach Using Lightweight CNNs for Significant ViewPoint and Appearance Changes," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 561–569, 2020, doi: 10.1109/TRO.2019.2956352.

[10] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," *Intell. Ind. Syst.*, vol. 1, no. 4, pp. 289–311, 2015, doi: 10.1007/s40903-015-0032-7.

[11] M. A. K. Niloy *et al.*, "Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review," *IEEE Access*, vol. 9, pp. 35338–35370, 2021, doi: 10.1109/ACCESS.2021.3062557.

[12] G. A. Acosta-Amaya, J. M. Cadavid-Jimenez, and J. A. Jimenez-Builes, "Three-Dimensional Location and Mapping Analysis in Mobile Robotics Based on Visual SLAM Methods," *J. Robot.*, vol. 2023, 2023, doi: 10.1155/2023/6630038.

[13] C. Liu, J. Xu, and F. Wang, "A Review of Keypoints' Detection and Feature Description in Image Registration," *Sci. Program.*, vol. 2021, 2021, doi: 10.1155/2021/8509164.

[14] M. Aladem and S. A. Rawashdeh, "Lightweight visual odometry for autonomous mobile robots," *Sensors (Switzerland)*, vol. 18, no. 9, pp. 1–14, 2018, doi: 10.3390/s18092837.

[15] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, pp. 1–22, 2019, doi: 10.1177/1729881419839596.

[16] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image Matching from Handcrafted to Deep Features: A Survey," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 23–79, 2021, doi: 10.1007/s11263-020-01359-2.

[17] F. Hidalgo, "Evaluation of Several Feature Detectors / Extractors on Underwater Images towards vSLAM," pp. 1–16, 2020, doi: 10.3390/s20154343.

[18] P. Adhikari, B. Roy, O. Sinkar, M. Gupta, and C. Ningthoujam, "Experimental Analysis of Feature-Based Image Registration Methods in Combination with Different Outlier Rejection Algorithms for Histopathological Images †," pp. 1–9, 2023.

[19] M. Wasala, H. Szolc, and T. Kryjak, "An Efficient Real-Time FPGA-Based ORB Feature Extraction for an UHD Video Stream for Embedded Visual SLAM," 2022.

[20] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-Free Local Feature Matching with Transformers," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 4, pp. 8918–8927, 2021, doi: 10.1109/CVPR46437.2021.00881.

[21] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "MAGSAC++, a fast, reliable and accurate robust estimator," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1301–1309, 2020, doi: 10.1109/CVPR42600.2020.00138.

[22] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2432–2443, 2017, doi: 10.1109/CVPR.2017.261.

[23] Z. Li and N. Snavely, "MegaDepth: Learning Single-View Depth Prediction from Internet Photos," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2041–2050, 2018, doi: 10.1109/CVPR.2018.00218.

[24] C. Riu, V. Nozick, and P. Monasse, "Automatic RANSAC by Likelihood Maximization," *Image Process. Line*, vol. 12, pp. 27–49, 2022, doi: 10.5201/ipol.2022.357.

[25] D. Barath, J. Noskova, and J. Matas, "Marginalizing Sample Consensus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8420–8432, 2022, doi: 10.1109/TPAMI.2021.3103562.

[26] D. Barath, M. Ivashechkin, and J. Matas, "Progressive NAPSAC: sampling from gradually growing neighborhoods," 2019, [Online]. Available: http://arxiv.org/abs/1906.02295

[27] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, "Kornia: An open source differentiable computer vision library for PyTorch," *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, pp. 3663–3672, 2020, doi: 10.1109/WACV45572.2020.9093363.