Combining Geometry Information with SuperPoint and SuperGlue for Visual Place Recognition

Yu Zhan
SUSTech EEE
12232151@mail.sustech.edu.cn
Division of labor: equal

Chengjie Zhang
SUSTech MAE
11911918@mail.sustech.edu.cn
Division of labor: equal

***Abstract*—Visual place recognition (VPR) refers to the task of determining whether two given images represent the same location. VPR plays a crucial role in the field of robotics for verifying loop closures in autonomous driving. Image matching is a common approach used to accomplish VPR tasks. In recent years, deep learning-based feature extraction and matching methods, such as SuperPoint and SuperGlue, have shown superior performance in image matching compared to traditional handcrafted features and matching techniques. We leveraged SuperPoint and SuperGlue for VPR tasks on the Oxford RobotCar dataset, achieving good results. However, we still faced challenges in overcoming variations in lighting conditions between day and night, large viewpoint changes, and interference from dynamic objects. To address these challenges, we proposed a method that incorporates point cloud information. This method involves filtering matching pairs at both the match stage and the geometric stage. We used an SVM classifier to fuse the number of matching pairs from image matching and the results of point cloud-based ICP registration. The results demonstrated that our approach effectively mitigated the performance degradation caused by the challenging scenarios in VPR tasks. By incorporating point cloud information, we were able to enhance the robustness and accuracy of the VPR system.**

***Index Terms*—Visual Place Recognition, Geometric Verification, Image Matching**

https://github.com/Chen-Jacker/EE5346-Visual-Place-Recognition-Project/

I. Introduction

A. Visual Place Recognition

The VPR (Visual Place Recognition) task is a very important task in computer vision and it has many applications, such as 3D reconstruction [1], loopback monitoring verification for SLAM [2] and robot repositioning [3]. The task is to determine the proximity of the two images given two images, using the similarity of the visual information of the two images to determine the proximity of the locations where the two images were taken. However, factors dominated by lighting changes, large viewpoint changes and dynamic objects can affect the similarity of the visual information of the image pairs and thus interfere with the results of VPR. For example, shots are taken at different moments in the same place: day and night.

B. VPR Through Image Matching

Thanks to the invariance of local features on illumination changes and affine transformation and rotation, local-feature-based matching patterns have received a lot of attention and are one of the important solutions for VPR tasks. The basic idea is to extract and match features first and then make decisions. However, methods based on such patterns, both traditional [4], [5] and deep learning-based [6], [7], focus too much on the similarity of local texture information, and are difficult to reject false negative for large viewpoint changes (Fig. 1) and illumination changes (Fig. 2) and false positive for different places with similar local textures (Fig 3, 4). Solely relying on the number of matches as an evaluation metric for VPR is insufficient to handle challenging environments.

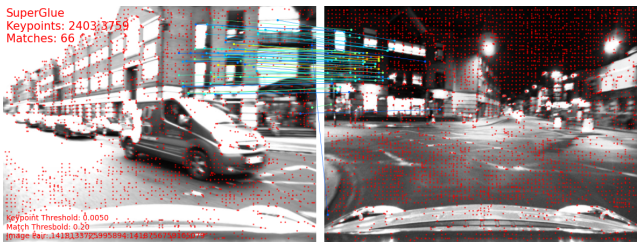

Fig. 1. Small IOU due to large viewpoint change results in a reduction in the number of matching points

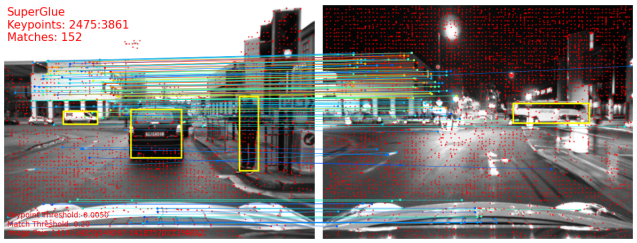

Fig. 2. Illumination changes affect the distribution of SuperPoint and reduce match pairs. Additionally, dynamic objects occlude texture-rich areas. Both lead to false negatives.

C. Our Method

We propose a series of improvements based on SuperPoint and SuperGlue to expand the feature dimensions beyond the number of matches. Specifically, these improvements are shown below:

- Adapt YOLOv8 to remove dynamic objects.
- Use RANSAC to improve 2D matching results.
- Use 3D matching pairs to generate geometric features.

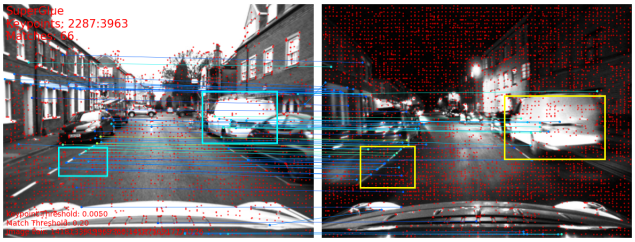

Fig. 3. Two different places with similar white van and road lines, which result in considerable matches that lead to false positive

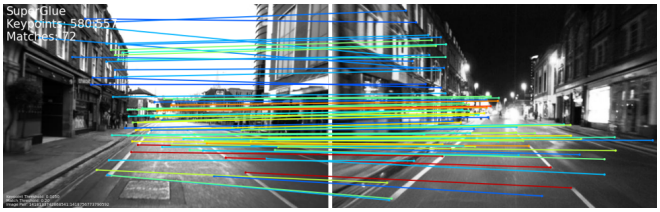

Fig. 4. Mismatching in different places with similar local textures.

Through these improvements, our method makes it possible to focus on local textures while fully taking into account the geometric information between image pairs, improving the overall effectiveness of VPR.

Since the course encourages the use of depth information, it was decided to use lidar point cloud from the Oxford RobotCar dataset. The addition of depth information makes the solution of the chi-square transformation matrix more accurate and restores the true scale of the displacement vector, making it possible to add a layer of geometric-level decision information.

D. Deviation from the Proposal

In our proposal, we planned to find an adaptive threshold adjustment method based on superPoint+superGlue and image lighting enhancement to deal with variations in illumination. However, this idea comes with a problem that it is too engineering and works on too small a problem. Considering that the course requires the use of an autonomous driving dataset, including the lidar data, the goal of our team shifted to designing a more general method to solve a VPR task in challenging autonomous driving environments.

II. Related Work

The traditional methods of this model are mainly based on handcraft's feature descriptors for matching and decision-making. Among them, SIFT+Ratio test [4] and ORB [5] + BruteForce Matcher are dominant. The former mainly uses SIFT [7] to extract keypoints and generate descriptors, and the ratio test will determine whether two keypoints can be used as matching pairs; the ORB of the latter mainly consists of the keypoint extractor Oriented FAST and the keypoint descriptor Rotated BRIEF, and then performs two-way brute force matching based on Hamming distance to use the pair of keypoints with the shortest distance as matching pair. The SIFT+Ratio test method has strong scale and rotation invariance and certain affine transformation invariance, which can produce a large number of matching pairs but is slow, while the ORB+BruteForce Matcher method has weaker scale and rotation invariance than the SIFT+Ratio test but is fast. Although both of them have certain lighting invariance, it is still difficult to cope with drastic lighting changes and large viewpoint changes and can be mismatched for different objects with similar local textures.

In recent years, deep learning-based feature matching methods have emerged and surpassed traditional methods in image matching problems. The representative methods are SuperPoint [8] + SuperGlue [7], which is an upgraded version of MagicPoint [9] and uses Homographic Adaptation for reliable self-supervised learning to improve the number and quality of key point monitoring; SuperGlue matches keypoints based on graph neural networks incorporating self-attention and cross-attention mechanisms. Both mechanisms are added to mimic the human image matching process, with the personal understanding that the self-attentive mechanism is similar to the keypoint descriptor while the cross-attentive mechanism is similar to performing matching. Fig. 5 shows the visualization results of self-attention and cross-attention.

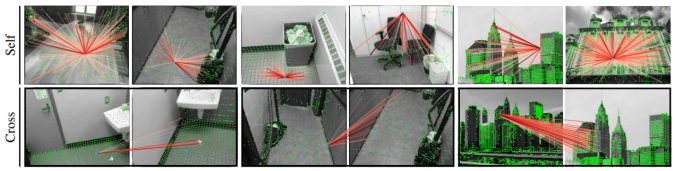

Figure 7: **Visualizing attention.** We show self- and cross-attention weights α_{ij} at various layers and heads. SuperGlue exhibits a diversity of patterns: it can focus on global or local context, self-similarities, distinctive features, or match candidates.

Fig. 5. The visualization results of self-attention and cross-attention.

The SuperPoint+SuperGlue strategy is able to obtain a large number and high quality of image matching pairs well, and the matching performance is significantly improved compared with the traditional method, so it can describe the similarity of two images more accurately and solve the VPR task better.

III. Methodology

We utilized the popular SuperPoint and SuperGlue methods in the field of image matching to match the test image pairs and obtain the number of matching pairs. To address the limitation mentioned in the introduction, where relying solely on the number of matching pairs is insufficient to handle challenging scenarios, we incorporated point cloud data to augment the feature representation dimension of a scene in a single frame image. Additionally, we mitigated the impact of dynamic objects and improve the 2D matching result by utilizing an object detection algorithm and RANSAC to filter them out respectively. We will then briefly describe the contents and functions of each module.

- **Preprocess**. We remove the car front cover that appears at a fixed position in each image to improve the effectiveness of image matching features on VPR.
- **Matching Stage**. Firstly, SuperPoint+SuperGlue is used to obtain a large number of matching pairs, then Yolov8

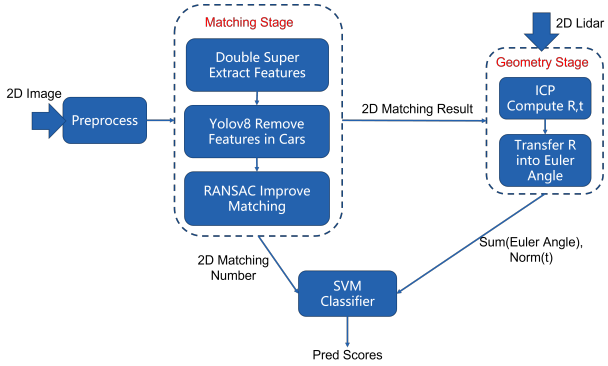

Fig. 6. The overview of our method. 'Double Super' means SuperPoint+SuperGlue, 'Sum(Euler Angles)' means add yaw, pitch and roll together and 'Norm(t)' means the modulus length of the displacement vector t

is used to eliminate the matching pairs that fall on dynamic objects (cars), and finally RANSAC is used to eliminate the matching pairs that do not satisfy the basic geometric relationship, and the remaining matching pairs are outputs.

- **Geometry Stage**. The point cloud around the image taken location is first generated based on the 2D radar data and then projected these point cloud into the 2D image, then find out the point cloud corresponding to the matching pair based on the image matching result, and compute the pose change (R, t) between two image frame using ICP. R is converted into Euler angles and compute the absolute value of each euler angle and then summed up as 'Sum(Euler Angles)'. Compute the modal length of t, which is the translating distance 'Norm(t)'.
- **SVM-based feature fusion**. Finally, we combine the output number of matching pairs, Sum(Euler Angles) and Norm(t) into a 3-dimensional feature vector, and then put it into a linear SVM for training, and output the prediction scores.

Next, we will describe each of the above-mentioned parts in detail.

A. Preprocess

Every image in the RobotCar dataset will have a car front cover, which will increase the number of meaningless matching pairs and thus decrease the effectiveness of the matching pair number feature on VPR, therefore, we remove the car front cover that appears in every image. We did not perform any stretching on the images after the removal, in order to ensure the normal shape. It is worth noting that all methods including baseline in the later comparison experiments are developed based on the images with the car front cover removed. Fig. 7 shows the removal results, (a) before removal and (b) after removal.

B. Matching Stage

We first use SuperPoint+SuperGlue to extract the feature points and matching pairs of the image pairs. Here, we set the maximum number of keypoints to 4096,

Fig. 7. (a) before removing car front cover and (b) after removing car front cover.

which is four times higher than the outdoor default of 1024 to obtain a larger number of matching pairs. The code of SuperPoint+SuperGlue is taken from the github source code made public by the author of SuperGlue (https://github.com/magicleap/SuperGluePretrainedNetwork). Then we use the advanced object detection network, Yolov8 [10], to monitor the small cars and buses in the pictures. After getting the detection box, as long as the matching pair exists with points falling in the detection box, then the matching pair is directly removed. Subsequently, we call opencv-python's RANSAC method of computing the essential matrix E based on the camera's intrinsic matrix to eliminate the unsuitable matching pairs. Finally, matching pairs that do not contain dynamic objects but also satisfy the essential geometric relationship are obtained. Fig. 8 shows the results in the matching stage. (a) is the result of using SuperPoint+SuperGlue output only, and (b) is the result of re-processing with Yolov8 and RANSAC.

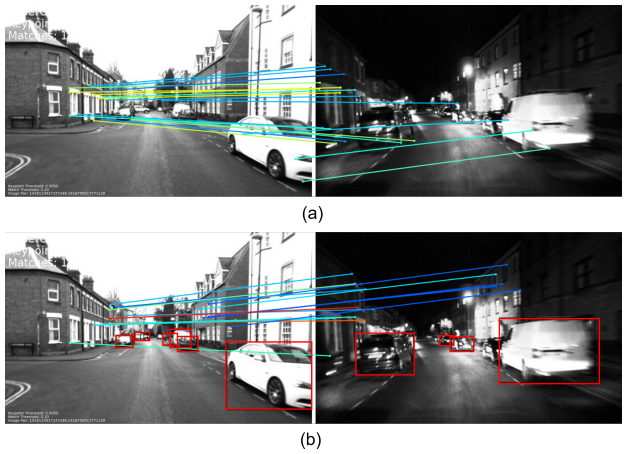

Fig. 8. (a) The result of using SuperPoint+SuperGlue output only, (b) The result of processing with Yolov8 and RANSAC.

C. Geometry Stage

1) Generate Point Cloud: Due to the limited vertical field of view (VFOV) of the provided 3D LiDAR in the dataset, which is only 3.2 degrees, the coverage range of the projected LiDAR point cloud in the image is too small. We have decided

to utilize data from a single-line 2D LiDAR. The 2D LiDAR in the dataset has a FOV of up to 270 degrees, fixed on the front of the vehicle and scanning vertically downwards, thereby capturing the point cloud distribution of the road surface and buildings on both sides.

Since a single frame of 2D LiDAR point cloud has a limited number of points, we generate a frame of point cloud by aggregating multiple 2D LiDAR scans within 20 meters ahead and behind the vehicle during its motion. The pose transformation of the vehicle required for the fusion of point clouds is provided by GPS and IMU data. The generated point cloud, as shown in the figure, provides denser information, which assists in finding the corresponding depth of the points in image matching pairs. Furthermore, this method avoids sampling dynamic objects in front and behind the vehicle, retaining only the point cloud information of the road surface and the 2 sides, significantly reducing the impact of dynamic objects on place recognition. The generated point cloud and a comparison with 3d point cloud are shown in Fig. 9.

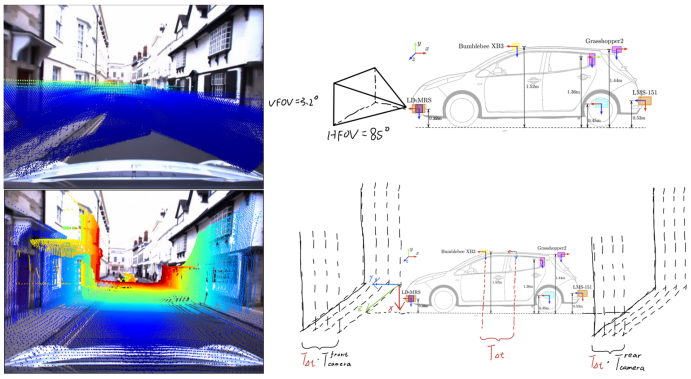

Fig. 9. The point cloud generated by aggregating 2D lidar scans is much more better than 3D lidar's direct output

2) Associate Matching Results with Point Cloud: We obtain the point cloud by the above method and project it onto the 2D image, and then find the corresponding point cloud of the 2D matching pair based on the obtained 2D image matching result. Due to the nature of the number and distribution of point clouds, not every pixel point on the image has a depth. In this regard, we have designed a new method to find the point cloud around the 3D position corresponding to the 2D matching pair. This method ensures that the found point cloud is not too far from the 3D position of the matching pair.

First, we use a multi-scale square box centered on the match point to get closer pixels with depth. The side length of the square box is gradually increased until a pixel with depth is found and then terminated. The range of edge length is from 3 to 51 pixels. It is important to note that it is possible to find more than one pixel with depth. We then determine whether the x and y theoretical maximum deviation of the acquired pixel's point cloud exceeds a threshold value of 0.2 m. The assumption of the theoretical maximum deviation is that the depths of the regions contained within the box (the representative depth of the box) are similar. Assuming that the pixel position x of the left endpoint of the box is u_l, the pixel position x of the right endpoint is u_r, and the representative depth of the box is d. The x of the spatial position corresponding to the pixel of the left endpoint is X_l, and the right endpoint corresponds to X_r with the intrinsic parameter f_x, then if the absolute value of $X_r - X_l$ has the upper limit X_{max}, then the judgment formula is below:

$$\frac{d|u_r - u_l|}{f_x} \leq X_{max}. \tag{1}$$

Also, the judgment about y is similar:

$$\frac{d|v_r - v_l|}{f_y} \leq Y_{max}. \tag{2}$$

For a match point that does not satisfy (1) or (2), we remove its corresponding match pair. We constrain the depth so that if the standard deviation of the depth corresponding to the searched pixel with depth is greater than 0.15, we discard the matching pair corresponding to this matching point. Fig. 10 shows the matched pairs with search results, black points are the matched points, and white points are the searched points with depth. Note that some white point positions may overlap with black points and be covered by black points, thus showing no white points near the black points.

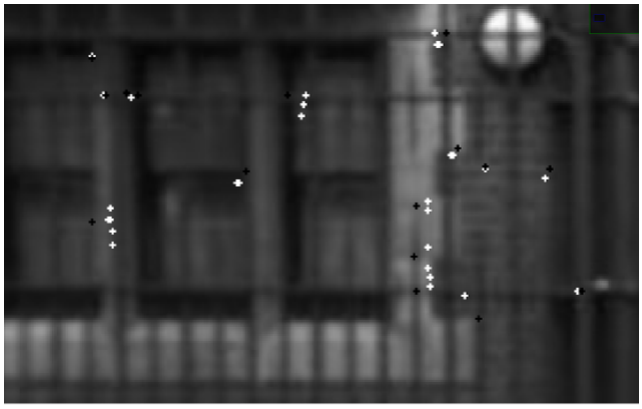

Fig. 10. The matched pairs with search results, black points are the matched points, and white points are the searched points with depth.

Finally, the point cloud for each matching point is obtained by averaging all the point clouds corresponding to the pixel points in the vicinity of each matching point.

3) Pose Estimation with Matched ICP: When the matching relationship between point clouds is given, there is an optimal analytical solution for ICP [11], which is mainly implemented by SVD. We write the algorithm by hand to obtain the rotation matrix R and the displacement vector t.

D. SVM-based Feature Fusion

We synthesize the number of matching pairs, Sum(Euler Angles) and Norm(t) into a 3-dimensional feature vector, then train and classify it using a linear SVM, and finally output the prediction scores. SVM is implemented by calling python's sklearn package.

E. Other Possible Method

The above-mentioned ICP method based on image matching pairs does not deviate from the constraint of image matching results. Therefore, we have decided to utilize the structural information of the scene point cloud to calculate the distance between the two images. Currently, we have two approaches in mind, but we have not yet obtained results.

The approach is to use downsampled point clouds for registration to compute the transformation between the two frames of images. This transformation can be considered as one of the constraints added to the scene feature vector. The second approach involves using point cloud descriptors. We plan to leverage the work of [12] that utilizes a deep learning network to output descriptors for single-frame point clouds. The pipeline of [12]'s work is shown in Fig. 11. It achieves 96.3 Average Recall@1 in the Oxford RobotCar dataset. We aim to calculate the distance between the point cloud descriptors corresponding to the two frames of images and include it as one of the constraints in the scene feature vector.

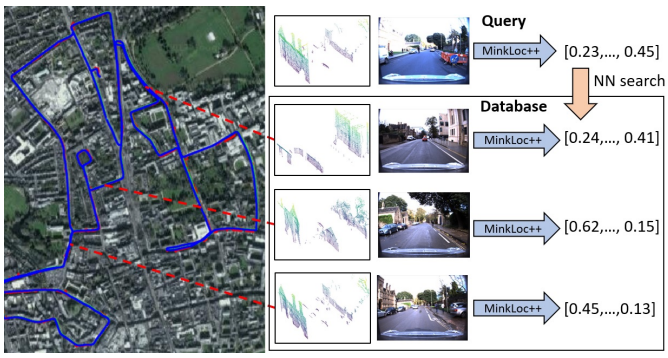

Fig. 11. MinkLoc++: Lidar and Monocular Image Fusion for Place Recognition

Regarding the first approach, due to the large-scale street scene and the fact that the point cloud shapes are mostly cylindrical when representing a street scene, the transformation obtained from the ICP algorithm differs significantly from the GPS and IMU values. Therefore, we have temporarily set aside this approach. As for the second approach, the work of [12] has a different data format from our work, requiring significant effort to reproduce. We have not completed this task yet. Additionally, the motivation for adopting this approach is still questionable since point cloud descriptor-based scene recognition methods are primarily used in the first stage of scene recognition rather than the verification stage (second stage).

IV. Experiment

A. Experiment Setting

We use SIFT+ratio test (sift), ORB+Brute force (orb), and SuperPoint+SuperGlue (pp) as our baselines. In order to prove the validity of each of our modules, we generated three methods using permutations, without Yolov8 and RANSAC (pp+icp), without RANSAC (pp+icp+yolo) and the full method (pp+icp+yolo+ransac). The evaluation datasets we use are dbNight easy, dbNight diff, dbSunCloud easy, and dbSunCloud diff, and the evaluation metrics used are F1 Score, Average Precision(AP), and recall under 100% precision (R@100P). To demonstrate the generalization ability of our model, we use our method to evaluate the dbNight diff and dbSunCloud diff datasets with the SVM models trained by dbNight easy and dbSunCloud easy, respectively.

B. Expreiment Results

The following Fig. 12-17 show the PR curves and comparison tables of the three metrics for the dbNight easy, dbNight diff, dbSunCloud easy and dbSunCloud diff datasets, respectively.

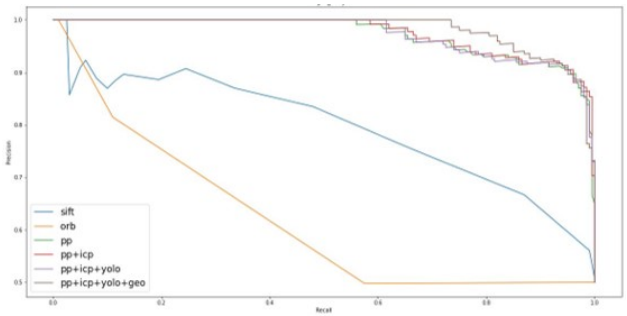

Fig. 12. PR curve of baselines and our methods under dbNight easy dataset.

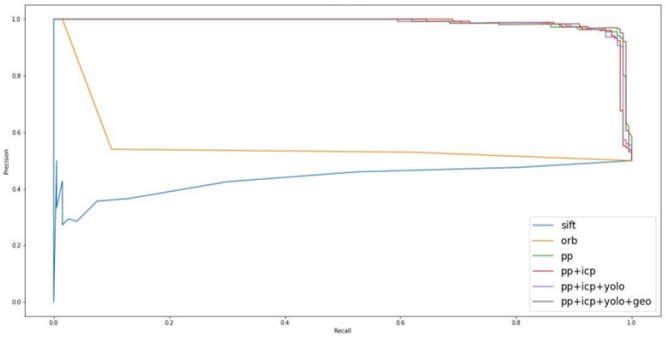

Fig. 13. PR curve of baselines and our methods under dbNight diff dataset.

V. Conclusion and Future Work

For the Visual Place Recognition task, we utilized the number of matching pairs from SuperPoint + SuperGlue image matching as an evaluation metric. Our test dataset was the Oxford RobotCar dataset. The results showed that, compared to traditional handcrafted descriptors (ORB, SIFT) and matching methods (Brute Force, ratio test), the SuperPoint + SuperGlue-based approach performed significantly better in terms of F1 Score, Average Precision (AP), and recall under 100

However, during our experiments, we found that the SuperPoint + SuperGlue approach still lacked invariance to three challenging scenarios: large viewpoint changes, illumination

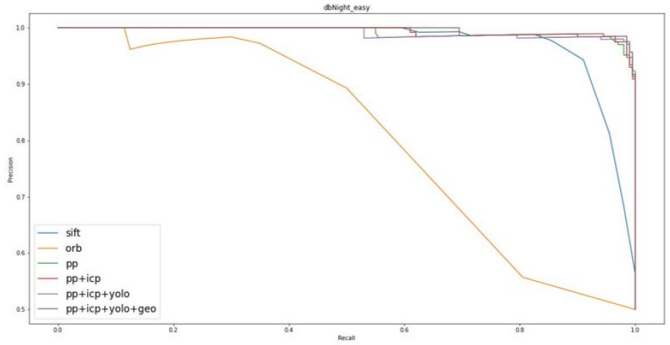

Fig. 14. PR curve of baselines and our methods under dbSunCloud easy dataset.

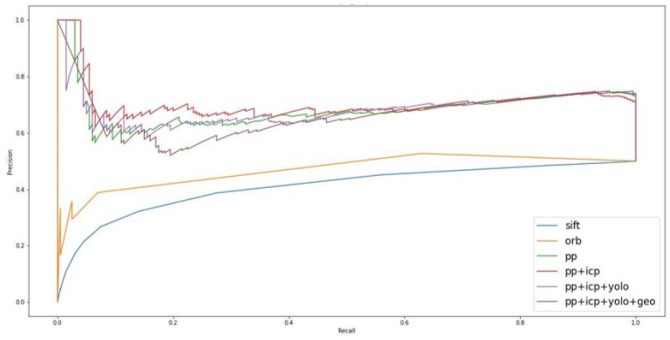

Fig. 15. PR curve of baselines and our methods under dbSunCloud diff dataset.

	SIFT	ORB	PP	PP+ICP	PP+ICP+Yolo	PP+ICP+Yolo+Geo
F1 Score	75.49	66.67	93.01	**93.20**	93.14	92.94
AP	76.72	53.55	97.18	**97.44**	97.23	98.17
R@100p	2.5	1.0	56.0	58.5	61.5	**73.5**

Fig. 16. Evaluation of baselines and our methods under dbNight easy dataset.

	SIFT	ORB	PP	PP+ICP	PP+ICP+Yolo	PP+ICP+Yolo+Geo
F1 Score	-	66.67	96.53	96.26	95.98	**97.27**
AP	45.33	52.65	98.80	98.53	98.58	**98.90**
R@100p	0.0	1.5	64.5	**69.0**	59.5	62.0

Fig. 17. Evaluation of baselines and our methods under dbNight diff dataset.

	SIFT	ORB	PP	PP+ICP	PP+ICP+Yolo	PP+ICP+Yolo+Geo
F1 Score	92.62	66.67	97.51	98.01	98.02	**98.50**
AP	96.87	74.56	99.21	99.37	99.18	**99.56**
R@100p	59.5	11.5	55.0	61.0	53.0	**69.5**

Fig. 18. Evaluation of baselines and our methods under dbSunCloud easy dataset.

	SIFT	ORB	PP	PP+ICP	PP+ICP+Yolo	PP+ICP+Yolo+Geo
F1 Score	-	-	84.93	83.58	85.11	**85.41**
AP	43.21	50.65	68.86	**70.87**	68.97	65.54
R@100p	0.0	0.0	3.0	**4.0**	1.5	0.0

Fig. 19. Evaluation of baselines and our methods under dbSunCloud diff dataset.

changes between day and night, and interference from dynamic objects.

So we conclude that due to the focus of SuperPoint on local texture similarity, while to some extent neglecting geometric topological information, solely relying on the number of matches as an evaluation metric for Visual Place Recognition is insufficient to handle challenging environments.

To address these limitations, we proposed two methods: one that utilizes object detection algorithms to remove feature points related to dynamic objects during the matching stage, and another that utilizes point cloud information for 2D match-based ICP during the geometric stage. These methods aim to enhance the invariance of dynamic objects and improve the ability to describe spatial geometric information. Finally, we improved the baseline SuperPoint + SuperGlue approach at the decision level by combining image matching results and point cloud geometric verification using an SVM classifier.

The results including the PR curve and three metrics showed improvements of our method, particularly in challenging scenarios with lighting variations between day and night, where our approach achieved great enhancements to improve R@100P from 56 to 73.5.

For future work, we consider integrating image matching information and point cloud descriptor at a more foundational level. In our proposed approach, the fusion of features is achieved by combining the scores from the image and geometric verification using an SVM classifier. However, this approach results in a relatively loose fusion of data between these two modalities, resembling a concatenation of two separate scoring modules, which may limit its performance.

In fact, the work of [12] has successfully fused image and point cloud descriptors at the feature level. We can draw inspiration from their approach and combine it with the latest advancements in image matching to achieve a more effective feature fusion.

Additionally, in our proposed match-based ICP method, the utilization of point cloud depth information for geometric verification is relatively simple. We need to address the challenge of effectively integrating the image matching points with their corresponding depth information. This is an aspect that we will consider in our future research.

REFERENCES

[1] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.-H. Liu, "Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2831–2840.

[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] T. Shan, B. Englot, F. Duarte, C. Ratti, and D. Rus, "Robust place recognition using an imaging lidar," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5469–5475.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[6] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[7] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[8] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[9] ——, "Toward geometric deep slam," *arXiv preprint arXiv:1707.07410*, 2017.

[10] J. Komorowski, "Improving point cloud based place recognition with ranking-based loss and large batch training," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 3699–3705.

[11] G. C. Sharp, S. W. Lee, and D. K. Wehe, "Icp registration using invariant features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 90–102, 2002.

[12] J. Komorowski, M. Wysoczańska, and T. Trzcinski, "Minkloc++: lidar and monocular image fusion for place recognition," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.