MMDF: Multi-Modal Deep Feature Based Place Recognition of Mobile Robots with Applications on Cross-scene Navigation

Xiang Yu, Bo Zhou, Zeqing Chang, Kun Qian and Fang Fang

***Abstract*—Although the navigation of robots in urban environments has achieved great performance, there is still a problem of insufficient robustness in cross-scene (ground, water surface) navigation applications. An intuitive idea is to introduce multi-modal complementary data to improve the robustness of the algorithms. Therefore, this paper presents an MMDF (multi-modal deep feature) based cross-scene place recognition framework, which consists of four kinds of modules: LiDAR module, image module, fusion module and NetVLAD module. 3D point clouds and images are input to the network firstly. The point cloud module uses PointNet to extract point cloud features. The image module uses a lightweight network to extract image features. The fusion module uses image semantic features to enhance point cloud features, and then the enhanced point cloud features are aggregated using NetVLAD to obtain the final enhanced descriptors. Extensive experiments on KITTI, Oxford RobotCar and USVInland datasets demonstrate MMDF outperforms PointNetVLAD, NetVLAD and a camera-LiDAR fused descriptor.**

***Index Terms*—Field Robots, Sensor Fusion, Deep Learning Methods**

I. Introduction

Perception and understanding of surroundings are indispensable for the navigation of mobile robots, while place recognition plays a vital role in this issue [1] [2]. The current place recognition methods are mainly designed for a single scene (such as ground, water surface and underwater) and only use single-modal information (single images or point clouds). Because different scenes have different environmental characteristics such as object types, lighting conditions and interference, cross-scene place recognition for the navigation of mobile robots still remains a challenging problem. For instance, place recognition methods [3] even fail to work properly in water surface scene, which has also been investigated in our previous work [4].

The main reason why the current methods cannnot solve the cross-scene place recognition is the limitation of single-modal perception. The images contain rich textures and semantic information, but are subject to variations in a viewpoint, illumination and weather. LiDAR point clouds contain sufficient geometric information, but lack descriptions of the appearance of objects and are less robust to changes in environmental types [4] [5]. Therefore, combining the two complementary modalities to conduct multi-modal fusion is an intuitive idea to solve the cross-scene place recognition.

However, the current mainstream place recognition methods simply concatenate the descriptors, which ignore the relationship between two modalities and lead to a waste of multi-modal information [6] [7]. A novel deep fusion place recognition method is more worthy of expectation for solving cross-scene applications. Therefore, in this paper, we establish the spatial correspondence between images and point clouds and propose a multi-modal deep feature based global descriptors for cross-scene place recognition.

The main contributions of this paper are as follows:

1) A multi-modal place recognition framework based on deep fusion is proposed for cross-scene application of place recognition. The comprehensive description of the place is constructed using image and point cloud feature descriptors. To the authors' knowledge, MMDF is the first multi-modal deep fusion feature for cross-scene place recognition. The multi-modal perception captures high-level semantic and geometric features in cross-scene applications, and is more robust to the long-term application of the robots.

2) A multi-modal fusion module for cross-scene place recognition is proposed. MMDF is able to establish the spatial correspondence between images and point clouds, and enhances the point cloud features by image semantic features without image annotations, so as to provide better environmental adaptability especially for environments with sparse buildings.

3) For the first time, our approach is focused on general purpose cross-scene place recognition and is able to apply place recognition methods on unmanned autonomous platforms like UGVs and USVs. Experiments on KITTI dataset, Oxford RobotCar dataset of ground scenes on UGVs and USVInland dataset of water surface scenes on USVs demonstrate that our multi-modal approach improves the robustness of place recognition for cross-scene applications.

II. Related works

A. Multi-modal Fusion

In the background of cross-scene application, we focus on the fusion of image and point cloud. Camera-LiDAR fusion is popular in 3D object detection which extract object-level

This paper was recommended for publication by Editor Pauline Pounds upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by supported by the National Natural Science Foundation (NNSF) of China under the Grants No. 62073075. *(Corresponding Author:Bo Zhou.)*

The authors are with School of Automation, Southeast University, Nanjing 210096, P. R. China (email: yuxiang@seu.edu.cn; zhoubo@seu.edu.cn; changzeqing@seu.edu.cn; kqian@seu.edu.cn; ffang@seu.edu.cn)

Digital Object Identifier (DOI): 10.1109/LRA.2022.3176731

features, while place recognition requires an overall description of the place, so that pipelines for 3D object detection cannot be directly applied to place recognition. According to the position, multi-modal fusion can be divided into three ways: early fusion, deep fusion and late fusion.

MV3D [8] and AVOD [9] are the pioneers of deep fusion. They convert LiDAR point clouds into BEV pseudo-images, and merge features of RGB image and BEV map directly. However, the BEV map inevitably loses a lot of useful information. In recent three years, many outstanding scholars have proposed many more elaborate methods. In recent three years, scholars have proposed many more elaborate methods. EPNet [10] uses feature extractors for both geometric and image stream and achieve the semantic level fusion of multi-scale information.

As to early fusion, PointPainting [11] first performs semantic segmentation on image, and appends the classification score of each projected point to the original. Finally, methods like PointPillars [12] and PointRCNN [13] are used for 3D object detection. PI-RCNN [14] includes two sub-networks for 2D semantic segmentation and point-based 3D detection. The author proposes a PACF module for fusing the results of image semantic segmentation and the proposals generated by 3D point clouds.

One representative pipeline of late fusion is CLOCs [15], which use the geometric and semantic consistency between 2D and 3D detection, and automatically learn probability dependence from training data for fusion.

B. Place Recognition

The LiDAR-based place recognition methods in recent years can be divided into global descriptor-based methods and segments-based methods.

The global descriptor-methods use deep neural network to generate global descriptor of input directly. M2DP [16] projects 3D point clouds onto a series of 2D planes, and use the density distribution of each point for optaining descriptors. PointNetVLAD [5] combines the point cloud feature extraction network PointNet [17] with feature aggregation network NetVLAD [18], and uses triplet loss to train the network. LPDNet [19] uses PointNet++ [20] to enhance the ability of the local feature extraction, learn the structure information of point cloud and reveal the spatial distribution of local features both in Feature space and Cartesian space.

The segments-based methods need to detect and match the key points or local areas in the point clouds. SegMatch [21] and SegMap [22] proposes a feature description method based on segmentation. They first segment the original point cloud, extract features from the segments, and then combine them in order to get the representation of the place.

Vision-based place recognition methods in recent years can be divided into global descriptor based methods and local patch descriptor based methods. For global descriptor based methods, NetVLAD [18] uses deep neural network to reimplement VLAD [23] that aggregate local descriptors to global descriptors. [24] uses joint 3D geometry and semantic information of the environments to build the global descriptor. For local patch descriptor-based methods, [25] proposes a self-supervise region-based method and uses image-to-region similarities enhance the learning of local features. [26] proposes a multi-scale fusion technique to generate and combine the locally-global descriptors of different sizes.

Although LiDAR-based place recognition has made great progress, the point clouds collected by the mobile robots are not easy to identify in cross-scene applications due to the limitations of water surface environment and platform, as shown in Fig. 6. Fortunately, the images taken on the water surface contain more recognizable environmental information. Therefore, multi-modal fusion for cross-scene place recognition become an intuitive idea.

Image and point cloud modal fusion place recognition methods are currently rare. [7] is one of the representatives, which uses PointNetVLAD [5] and ResNet [27] to extract the feature of point cloud and image respectively, and merge them into a global feature descriptor. However, it is simply a relatively isolated fusion, and ignore the intrinsic relationship between two modals. [28] builds an end-to-end pipeline of place recognition based on spherical projection of cameras and LiDARs. [4] presents a robust cross-scene SLAM algorithm for ground and surface applications.

Therefore, we proposed a point-based multi-modal deep fusion algorithm that augments each point features with corresponding image features. Our research is pioneering for cross-scene place recognition.

III. Multi-modal deep feature based place recognition

Assuming that two scenes are M_1 and M_2, a mapping $f(\cdot)$ needs to be found so that $f(M_1, M_2)$ can be used to measure the similarity, as in

$$f(M_1, M_2) = \|g(M_1) - g(M_2)\|_2, \tag{1}$$

when $f(M_1, M_2)$ is larger, it means that the distance between $g(M_1)$ and $g(M_2)$ is farther, that is, M_1 and M_2 are less similar; When $f(M_1, M_2)$ is smaller, it means $g(M_1)$ and $g(M_2)$ is closer, that is, M_1 and M_2 are more similar.

The whole end-to-end network consists of four modules shown in Fig. 1: LiDAR module, image module, fusion module and NetVLAD module. The fusion module transforms image features into point-wise features to enhance point cloud features in different dimensions. Because image features are susceptible to light and seasonal changes, weight map is generated for point-wise image feature to adjust the weight of image features. This deep fusion scheme establishes the correspondence between point clouds and images, and fuses these features together to obtain a more overall representation.

A. LiDAR Module

We choose the most classic point-based method PointNet [17] as our LiDAR backbone, because point-based method is convenient to find the correspondence between the point in the

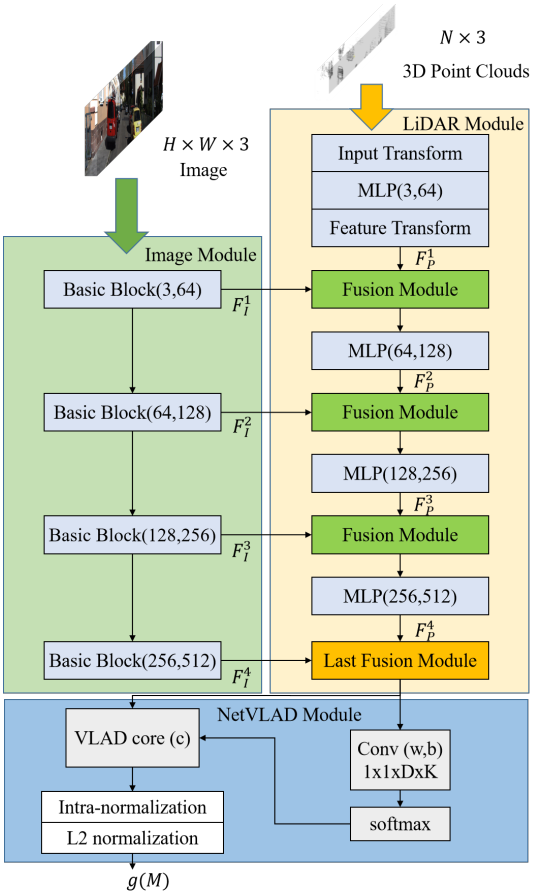


Fig. 1. Detailed architecture of our network. The MMDF network includes four modules: LiDAR module, image module, fusion module and NetVLAD module. LiDAR module uses PointNet to extract point cloud features. Image module constructs a lightweight network to extract image feature. Fusion module enhances the point cloud features with image features. The details of Fusion Module are indicated in Fig. 2. NetVLAD module is used to aggregate local features to place descriptors.

point cloud and the pixel in the image while the voxel-based method may lose some original spatial position relationship.

In order to perform feature fusion with the image module, PointNet is only used to extract the feature of the objects. The inputs of LiDAR module are the 3D point clouds and point-wise image feature, and the outputs are $F_p^i(i = 1, 2, 3, 4)$ representing the features in different scales in Fig. 1. $F_p^i(i = 1, 2, 3, 4)$ and point-wise image features are fused in four dimensions before being fed to next layer.

B. Image Module

The image network is composed of four lightweight basic convolution modules. Each basic module includes a 3×3 convolution layer, followed by a batch normalization layer and a ReLU activation function.So that the height and width of the feature maps can be kept constant after one basic convolution modules. The inputs of image module are RGB images without annotations, and the outputs are $F_I^i(i = 1, 2, 3, 4)$ which represent the feature maps in different scales in Fig. 1.

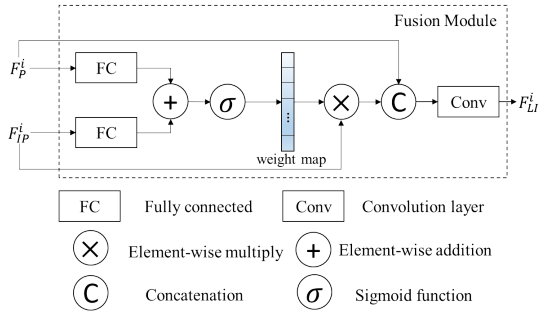


Fig. 2. Illustration of fusion module. First, point cloud features and point-wise image features are added after fully connected layer. Then, sigmoid function is used to generate a weight map for point-wise image feature. At last, the point-wise image feature is combined with point cloud feature. Except the last fusion module, other fusion modules need to be convoluted to the half of original dimension.

C. Fusion Module

The LiDAR-guided image fusion module shown in Fig. 2 can be summarized as three steps: spatial corresponding, point-wise image feature extraction, and feature fusion. This fusion module is inspired by PointPainting [11], and we propose the fusion module to fuse the image and LiDAR feature in the feedforward feature extraction process.

Firstly, The points in LiDAR coordinae system are converted to image coordinate system. For each frame including point clouds and image, a transformation matrix between different sensors can be generated.

Subsequently, $F_I^i(i = 1, 2, 3, 4)$ are transformed to point-wise image feature $F_{IP}^i(i = 1, 2, 3, 4)$ according to the corresponding value of each point in feature maps. We use bilinear interpolation to solve the problem that projected points may fall between the pixels and output the point-wise image feature. Because each point in the input point clouds gets a weight value in the feature map, the dimension of $F_{IP}^i(i = 1, 2, 3, 4)$ (64, 128, 256 and 512 in order) is the same as the feature dimension of $F_P^i(i = 1, 2, 3, 4)$.

Ultimately, $F_P^i(i = 1, 2, 3, 4)$ and $F_{IP}^i(i = 1, 2, 3, 4)$ are fused to $F_{LI}^i(i = 1, 2, 3, 4)$ by a convolution layer followed by a batch normalization and a ReLU activation and then concatenated together. Before feature fusion, we use a sigmoid function to generate weight maps for point-wise image feature F_{IP}^1, because it is extremely important to set weight for point-wise image features.

To generate weight maps, firstly F_P^1 and F_{IP}^1 are fed into fully connected layer to reduce the dimension from $N \times D(D = 64, 128, 256, 512)$ to $N \times 1$. Then the two $N \times 1$ point feature and image feature are added to obtain a feature intensity representation, which is then input to the sigmoid function to generate the $N \times 1$ weight map $\mathbf{w}$.

$$\mathbf{w} = \sigma(\mu F_P^1 + \nu F_{IP}^1) \tag{2}$$

where μ and ν denote the learnable weight matrices in the Fusion module, and σ denotes the sigmoid function. Finally, multiply the weight map and point-wise image feature together

and we get the modified F_{IP}^1 to concatenate with F_P^1, which can be formularized as:

$$F_{LI}^1 = F_P^1 || \mathbf{w} F_{IP}^1 \tag{3}$$

So that F_P^1 and F_{IP}^1 can be fused to a global feature to input to next convolution layer.

D. NetVLAD Module

After the last fusion module, the extracted fusion features only represent the local features. It is necessary to use a feature aggregation network to aggregate the local features into global features. We use the NetVLAD in PointNetVLAD [5], and the cluster size is 64. The input of NetVLAD module is a N 1024-dimensional feature vector per map frame, the global fused descriptor is a 128-dimensional vector.

E. Training Loss

Metric learning is a spatial mapping method that learns from an embedding space in which all data is transformed into a feature vector, and the feature vectors of similar samples have small distances from each other.

We use the contrastive loss [29] in metric learning to train the whole network, which can be described as

$$L_{cmp} = yd^2 + (1-y)\max(m-d, 0)^2 \tag{4}$$

where $d = |g(M_1) - g(M_2)|_2$ denotes the distance in the global fusion feature space between sample pair, m denotes the margin of distance, and y denotes the input label (0/1). When $label = 1$, the sample pair is positive, and $label = 0$ means the sample pair is negative. When $y = 0$ and $0 < d < m$, $L_{cmp} = \max(m-d, 0)^2 = (m-d)^2$, the L_{cmp} increases as the d decreases, which means the parameter d of negative pairs are training to increase to m. If d falls outside of m, the loss is 0, and the distance does not need to be optimized. When $y = 1$, $L_{cmp} = d^2$, the L_{cmp} increases as the d increases, which means the parameter d of positive pairs are training to decrease to 0. Thus, the parameter d at same position is small, while the d at different position is large.

F. Place Matching

For place matching, we use the Euclidean distance between the descriptors to judge whether the two frames of data belong to the same place. When the distance is lower than a certain threshold (25 in this paper), it is considered that the two frames are the same place.

IV. Experiments

A. Implementation Details

The purpose of our place recognition task is to complete the loop closure detection in SLAM. We use the precision-recall curve to evaluate the models instead of @N recall, which means that for each map frame used for query, it is sorted according to the confidence of prediction, and the top N results are taken for statistics and the recall rate is calculated.

To evaluate the performance of MMDF, we compare MMDF with PointNetVLAD [5], NetVLAD [18], the camera-LiDAR fused descritor (called CLFD in the evaluation) proposed in [7] and MMLF (ours multi-modal late fusion version). We fine tune the pretraining model from PointNetVLAD and NetVLAD under KITTI odometry dataset [30], and use them as the test model. MMDF, MMLF and CLFD share identical training sets. Moreover, MMLF is a late fusion of LiDAR module and Image module. The pure image descriptors output by image module and the pure point cloud descriptors output by PointNetVLAD are directly concatenated to obtain the global fused descriptors.

TABLE I
THE NUMBER OF TRAINING AND TEST PAIRS OF KITTI ODOMETRY DATASET

Set	00	02	05	06	07	09	Total
Train	948	566	440	534	2200	2968	9960
Test	3792	2270	1760	2138	0	0	7656
Total	4740	2836	2200	2672	2200	2968	

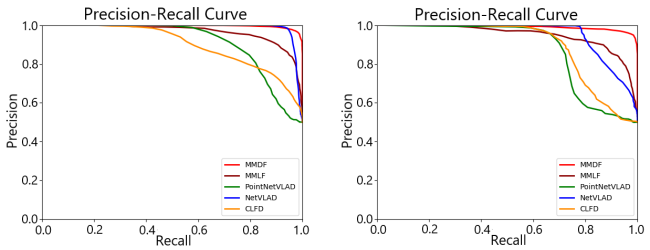
Point clouds are first preprocessed by filtering the points outside the image in image coordinate system by the method [31]. Then select 5000 points randomly as input. All the works are implemented based on Pytorch and SGD optimizer with a learning rate of 5×10^{-6} is used for training. The whole network is trained with the batch size of 8. We select the model having best performance under validation tests (generated from train sets) after 100 epochs. All the experiments are carried on Intel core i9-7900x and Nvidia GTX-1080 Ti.

B. Evaluation on KITTI odometry dataset

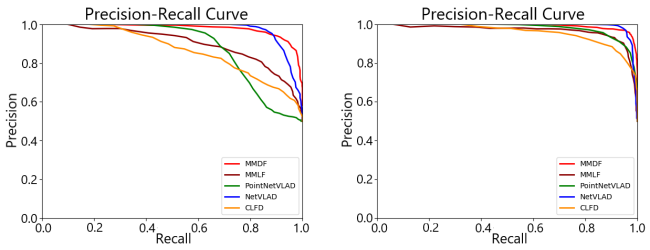
By experiments on KITTI Odometry dataset, the place recognition performance on general urban environments without drastic environmental changes can be evaluated. Sequence 00, 02, 05 and 06 are selected as the test sets. We build training set and test set by constructing positive and negative sample pairs. Two frames with true distance less than 5m are constructed as positive pairs, and two frames with true distance greater than 50m are constructed as negative pairs. The ground truth come from KITTI odometry dataset marks the distance between the corresponding two moments in the real world. We divide the corresponding frames of the loop closure into first-pass frames and second-pass frames. For each second-pass frame, find x ($x_{max} = 5$) frames in the first-pass frames to form x positive pairs, and then select x frames outside of 50m to form x negative pairs. So that the ratio of positive and negative pairs is $1:1$. Table I shows the number of KITTI training and test set. All generated pairs are divided into training and test sets by $1:4$. In order to increase the number of training sets, we select some adjacent frames (not from loop closure) in sequence 07 and 09 as positive pairs, and then build the same number of negative pairs.

We use the precision-recall curve to analyze the performance of several methods, and calculate the maximum F_1 score to evaluate the curve. F_1 score is defined as

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{5}$$



(a) KITTI 00 (b) KITTI 02



(c) KITTI 05 (d) KITTI 06

Fig. 3. Precision-recall curve on KITTI odometry dataset for general urban/country scenes evaluation. Our mean F1 max score exceeds other methods and the overall performance of precision-recall curve is better than them especially on sequence 02.

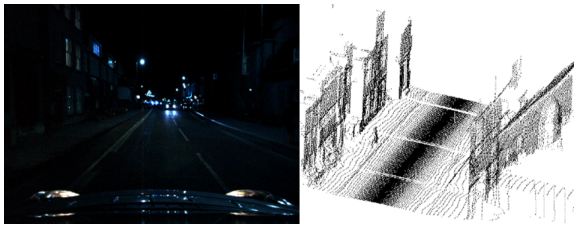
TABLE II
MAXIMUM F_1 SCORES ON KITTI ODOMETRY DATASET

Methods	Maximum F_1 Scores				Mean
	00	02	05	06	
PointNetVLAD [5]	0.820	0.791	0.779	0.920	0.828
NetVLAD [18]	0.963	0.874	0.916	0.962	0.929
MMLF(ours, late fusion)	0.903	0.884	0.824	0.924	0.884
MMDF(ours, deep fusion)	**0.973**	**0.967**	**0.927**	**0.963**	**0.958**
CLFD [7]	0.811	0.800	0.770	0.894	0.819

where P denotes precision and R denotes recall. It can be seen from the TABLE II that our average F_1 max score exceeds other four methods especially on sequence 02. However, on sequence 05, the performance of MMDF is slightly better than NetVLAD. The sequence 02 is the urban/country composite road scene with sparse buildings. Meanwhile, the sequences 00, 05 and 06 are all collected in the urban. The buildings are dense in urban area, and the features contained in the images and point clouds are more abundant. It makes the multi-modal fusion methods more robust when facing places of sparse features, as shown in Fig. 3 (b).

C. Evaluation on Oxford RobotCar dataset

Oxford RobotCar dataset [32] contains over 100 repetitions of a consistent route, which are in different light and weather, including night, sunny day, rainy day and other difficult scenes. The challenging lighting changes scenes (2014/11/14 and 2014/11/18, indicated in Fig. 4) are used for training and testing. All positive and negative pairs are generated in the same way as on KITTI Odometry dataset. The test set contains about 70k pairs.



(a) A sample frame from 2014/11/14(night).

(b) A sample frame of 2014/11/18(sun).

Fig. 4. Two frames of image and point cloud at same place from Oxford RobotCar dataset.

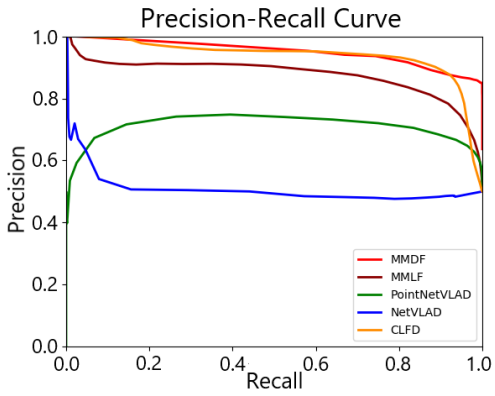


Fig. 5. The precision-recall curve for challenging lighting changes evaluation generated by the 2014/11/14 and 2014/11/18 sub-datasets.

TABLE III
MAXIMUM F_1 SCORES ON OXFORD ROBOTCAR DATASET

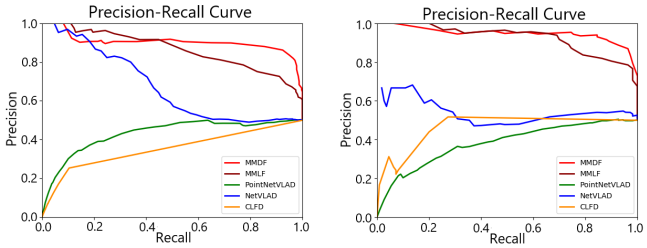
Methods	Maximum F_1 Scores
PointNetVLAD [5]	0.779
NetVLAD [18]	0.667
MMLF(ours, late fusion)	0.845
MMDF(ours, deep fusion)	**0.919**
CLFD [7]	0.899

The precision-recall curve is shown in Fig. 5 and the F1 max scores are indicated in TABLE III. NetVLAD using image-only methods is difficult to distinguish places when facing lighting change. PointNetVLAD, MMLF, MMDF and CLFD are relatively stable because point clouds are not affected by lighting change. MMDF performs more robust than the single-modal method after multi-modal fusion when facing severe illumination change, but is close to CLFD.

(a) A sample frame from H05_9.

(b) A sample frame of N03_2.

Fig. 6. Two frames data from USVInland dataset. It can be seen that there is a great difference between the water surface and the land environment. Compared with the land environment, the water surface environment is more open.



(a) H05_9. (b) N03_2.

Fig. 7. The precision-recall curve on sequence H05_9 and N03_2 for water surface scenes evaluation. Our MMDF improves the performance on N03_2 (overcast) and H05_9 (mist) by a wide margin.

TABLE IV
MAXIMUM F_1 SCORES ON USVINLAND DATASET

Methods	Maximum F_1 Scores		Mean
	H05_9(USV)	N03_2(USV)	
PointNetVLAD [5]	0.669	0.669	0.669
NetVLAD [18]	0.669	0.695	0.682
MMLF(ours, late fusion)	0.808	0.865	0.837
MMDF(ours, deep fusion)	**0.892**	**0.914**	**0.903**
CLFD [7]	0.667	0.667	0.667

D. Evaluation on USVInland overwater dataset

In order to test the performance of our method in water surface environments, we selected USVInland [3] dataset, which is a multi-sensor dataset for USVs in inland waterways. As Fig. 6 shows, the obtained point cloud in this dataset is much less than that on land while the main information in the image is water surface. The USVInland train set includes 2536 pairs in H05_9 and 888 pairs in N03_2 , while the test set includes 1585 pairs in H05_9 and 554 pairs in N03_2.

The precision-recall curves of four methods in two sequences are indicated in Fig. 7. In addition, F1 max scores of them are shown in TABLE IV. Because the point clouds on the water are too sparse, the obtained geometric features cannot completely represent the place. The PointNetVLAD has no ability to represent a place because of the poor point information. It can be seen from Fig. 6 that the water surfaces in image also limit the performance of NetVLAD. Unfortunately, after 100 epochs, CLFD still cannnot distinguish the places in the water surface. The possible reason is that L2 normalization is used in CLFD [7] to balance the features of the two modalities, and no weight map is set for a certain feature to select a better feature representation. MMDF improves the performance on N03_2 (overcast) by a wide margin, but on H05_9 (mist) is not obvious in Fig. 7. Fig. 8 indicates a negative sample pair in N03_2. It indicates that the distribution of point clouds in red and black color is hard to recognize. And the water surface also occupies most area of the image, which lead to a hard recognition for the algorithms.

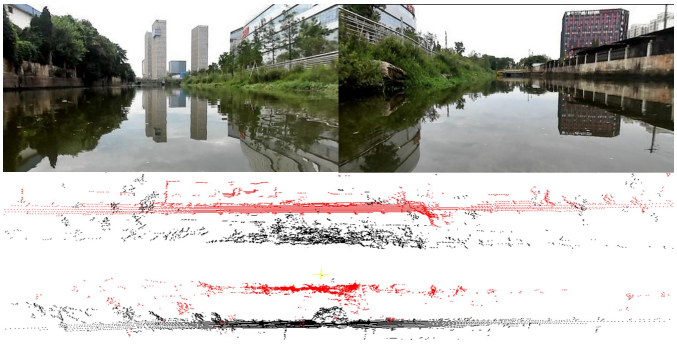


Fig. 8. A negative pair in sequence N03_2. The left figure and black points denote a frame, right figure and right points denote another frame. PointNetVLAD and NetVLAD mistakenly identify the negative pair as positive pair, but our method does not mistakenly identify them.

E. Efficency

For each frame, the size of our descriptor is 128. With the batch size of 16, the time costs of generating descriptors are 410ms (CLFD [7]), 28ms (MMDF and MMLF, ours), 22ms (PointNetVLAD [5]) and 6ms (NetVLAD [18]).

V. CONCLUSION

A multi-modal deep fusion feature MMDF for cross-scene place recognition is proposed in this paper. Our deep fusion applies the corresponding relationship between points in point clouds and pixels in images in the world coordinate system, and enhances the point clouds features with image features without image annotations. Although we have implemented deep fusion to build descriptors for the cross-scene place recognition, in order for the two modalities to be processed by the network, the number of point clouds input must be constant and each point needs an image pixel to correspond to it, which means that the effective area of the sensors is limited to the overlap area of the two sensors. Therefore, we will consider further utilizing point clouds and images from non-overlapped areas to build a more comprehensive representation of a place. In addition, we will conduct experiments on actual cross-scene environments to verify and improve our method.

REFERENCES

[1] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19516–19547, 2021.

[2] S. Arshad and G.-W. Kim, "Role of deep learning in loop closure detection for visual and lidar slam: A survey," *Sensors*, vol. 21, no. 4, p. 1243, 2021.

[3] Y. Cheng, M. Jiang, J. Zhu, and Y. Liu, "Are we ready for unmanned surface vehicles in inland waterways? the usvinland multisensor dataset and benchmark," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3964–3970, 2021.

[4] B. Zhou, Y. He, K. Qian, X. Ma, and X. Li, "S4-slam: A real-time 3d lidar slam system for ground/watersurface multi-scene outdoor applications," *Autonomous Robots*, vol. 45, no. 1, pp. 77–98, 2021.

[5] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4470–4479, 2018.

[6] A. Oertel, T. Cieslewski, and D. Scaramuzza, "Augmenting visual place recognition with structural cues," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5534–5541, 2020.

[7] S. Xie, C. Pan, Y. Peng, K. Liu, and S. Ying, "Large-scale place recognition based on camera-lidar fused descriptor," *Sensors*, vol. 20, no. 10, p. 2870, 2020.

[8] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

[9] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, IEEE, 2018.

[10] T. Huang, Z. Liu, X. Chen, and X. Bai, "Epnet: Enhancing point features with image semantics for 3d object detection," in *European Conference on Computer Vision*, pp. 35–52, Springer, 2020.

[11] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4604–4612, 2020.

[12] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

[13] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 770–779, 2019.

[14] L. Xie, C. Xiang, Z. Yu, G. Xu, Z. Yang, D. Cai, and X. He, "Pi-rcnn: An efficient multi-sensor 3d object detector with point-based attentive cont-conv fusion module," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 12460–12467, 2020.

[15] S. Pang, D. Morris, and H. Radha, "Clocs: Camera-lidar object candidates fusion for 3d object detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10386–10393, IEEE, 2020.

[16] L. He, X. Wang, and H. Zhang, "M2dp: A novel 3d point cloud descriptor and its application in loop closure detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 231–237, IEEE, 2016.

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

[18] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307, 2016.

[19] 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*, pp. 2831–2840, 2019.

[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[21] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5266–5272, IEEE, 2017.

[22] R. Dube, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "Segmap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, 2020.

[23] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3304–3311, IEEE, 2010.

[24] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic visual localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6896–6906, 2018.

[25] Y. Ge, H. Wang, F. Zhu, R. Zhao, and H. Li, "Self-supervising fine-grained region similarities for large-scale image localization," in *European Conference on Computer Vision*, pp. 369–386, Springer, 2020.

[26] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14141–14152, 2021.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[28] L. Bernreiter, L. Ott, J. Nieto, R. Siegwart, and C. Cadena, "Spherical multi-modal place recognition for heterogeneous sensor systems," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1743–1750, IEEE, 2021.

[29] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *2016 23rd international conference on pattern recognition (ICPR)*, pp. 378–383, IEEE, 2016.

[30] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361, IEEE, 2012.

[31] X. Meng, N. Currit, and K. Zhao, "Ground filtering algorithms for airborne lidar data: A review of critical issues," *Remote Sensing*, vol. 2, no. 3, pp. 833–860, 2010.

[32] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.