

Received May 14, 2019, accepted June 5, 2019, date of publication June 10, 2019, date of current version June 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921828

# A Hierarchical Attention Fused Descriptor for 3D Point Matching

**WENJUN SHI<sup>ID</sup><sup>1,2</sup>, DONGCHEN ZHU<sup>1</sup>, LIANG DU<sup>ID</sup><sup>1,2</sup>, GUANGHUI ZHANG<sup>ID</sup><sup>1,2</sup>, JIAMAO LI<sup>ID</sup><sup>1,2</sup>, (Member, IEEE), AND XIAOLIN ZHANG<sup>1,2,3</sup>, (Member, IEEE)**

<sup>1</sup>Bionic Vision System Laboratory, State Key Laboratory of Transducer Technology, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China

<sup>2</sup>School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

<sup>3</sup>School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China

Corresponding author: Jiamao Li (jmli@mail.sim.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671014, in part by the Shanghai Municipal Science and Technology Major Project (ZHANGJIANG LAB) under Grant 2018SHZDZX01, and in part by the Shanghai Science and Technology Committee, China, under Grant 16JC1420503.

**ABSTRACT** Motivated by recent successes on learning 3D feature representations, we present a Siamese network to generate representative 3D descriptors for 3D point matching in point cloud registration. Our system, dubbed HAF-Net, consists of feature extraction module, hierarchical feature reweighting and recalibration module (HRR), as well as feature aggregation and compression module. The HRR module is proposed to adaptively integrate multi-level features through learning, acting as a hierarchical attention fusion mechanism. The learnable feature pooling technique VLAD is extended into our aggregation module, which is further utilized to extract principal components of features and compress them into a low dimensional feature vector. To train our model, we amass a large dataset for 3D point matching. The dataset is composed of matched and unmatched point block pairs, which are automatically searched from existing reconstruction datasets with known poses. The experiments demonstrate that the proposed HAF-Net not only outperforms other state-of-the-art approaches in 3D feature representation but also has a good generalization ability in various tasks and datasets.

**INDEX TERMS** 3D descriptor, hierarchical attention, data reweighting, feature recalibration, aggregation, data driven, 3D point matching.

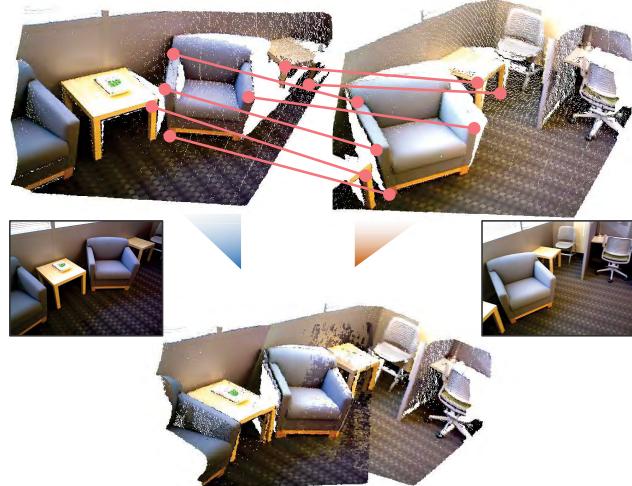
## I. INTRODUCTION

With the rapid development of artificial intelligence, robots are going to play very important roles in our daily life, such as autonomous driving, searching and rescuing, and industrial inspection. One fundamental component is to perceive the surroundings (known as scene perception) when a robot carries out its tasks in above-mentioned scenes. The 3D scene reconstruction is extremely important in the scene perception. Registering scene fragments from different views is a vital part of all the reconstruction methods for various forms of 3D data like the *multi-view image*, *volume*, *mesh*, and *point cloud*. Among them, point cloud is becoming increasingly popular as a standard 3D acquisition format used by range-scanning devices, such as LiDARs and

The associate editor coordinating the review of this manuscript and approving it for publication was Shuhan Shen.

RGB-D cameras. It is particularly amenable to geometric operations and can represent geometric details with little memory. Therefore, available 3D reconstruction is generally based on the point cloud data and many works are devoted to the point cloud registration.

Contemporary algorithms for point cloud registration can be categorized into two types. One is based on the iterative optimization of all points, like ICP [2] which attempts to minimize the distance function of two point sets with iterative optimization to obtain the transformation between them. The other relies on 3D keypoint matching. This kind of methods are not constrained by the range of motion between two fragments, and generally provide initial transformation for the iterative method. 3D points matching acting as a significant stage of unstructured point clouds registration remains an open research problem. Because current 3D features are not as robust as numerous stable 2D features, such as SIFT [3]



**FIGURE 1.** Our data-driven HAF-Net describes the input point blocks discriminatively to achieve correspondences matched from the given fragment pairs. This figure gives an example of registering two point clouds based on our proposed descriptors. The illustrative images and point clouds are taken from SUN3D [1].

and SURF [4], which have been widely utilized in the 2D keypoint matching of images. Researchers found it is difficult to generate 3D feature descriptors on the basis of certain rules for 3D scan data such as point clouds, just like the way in 2D images.

Recently, inspired by the successes of deep neural networks in the image field, some researchers try to design novel networks for 3D tasks. Some networks are emerged to directly process raw point clouds, such as PointNet [5]. Few of these methods are used for the 3D point matching, because it is difficult to obtain valid datasets for the 3D matching task. However, currently most deep models concentrate on how to better extend powerful representations merely from point coordinates  $xyz$  from point clouds, which illustrate that 3D coordinates are enough to describe local geometric information. Actually, in 2D images, complementary information such as intensity gradient information, edge information and so on will be combined with color information in order to get better feature descriptions. It is necessary to incorporate the complementary information of point clouds to form better 3D feature descriptors. Some models, such as PPFNet [6] and PPF-FoldNet [7], directly use  $xyz$  and *normal* information as a whole input. The 3D feature descriptors generated from these methods are not representative enough because they ignore the differences of the value level and data distribution between  $xyz$  and *normal* resulting from the physical significance of themselves. It is necessary to fuse different kinds of information efficiently, and utilizing attention mechanism to reweight, recalibrate and fuse different kinds of features is a reasonable solution. Therefore, in this paper we try to handle this problem through designing hierarchical attention structure for the effective fusion of different information.

In this paper, a learned 3D descriptor is proposed to match 3D points for point cloud registration. Different from 3DMatch [8], a network for testing the similarities of 3D TDF (Truncated Distance Function) batches, our novel HAF-Net (Hierarchical Attention Fused Network) can directly process point cloud data based on the PointNet [5]. For felicitously combining the hierarchical attention of the multi-type input features and the channel-wise features, a feature reweighting and recalibration module named HRR is designed. We also adopt a descriptor pooling method VLAD [9] (Vector of Locally Aggregated Descriptors), which is popular in the image field, to replace the max pooling layer in PointNet for feature aggregation. To summarize, our contributions are the following:

- The HAF-Net is designed to generate representative 3D descriptors for 3D point matching and point cloud registration. The proposed HAF-Net outperforms other state-of-the-art 3D representation methods in the 3DMatch benchmark [8].
- A novel module named HRR is proposed to fuse the attention of hierarchical features, which reweights multi-type information and recalibrates channel-wise features. The VLAD is first introduced to the 3D point matching task achieving the aggregation of fused-features effectively.
- A new dataset for point cloud classification and 3D point matching is amassed, which contains various simple basic point cloud structures, such as planes, wall corners, table corners, etc.

## II. RELATED WORK

### A. TRADITIONAL HAND-CRAFTED 3D FEATURE DESCRIPTORS

Like features in 2D images, 3D feature descriptors have also drawn increasing attention in recent years. Contemporary hand-crafted 3D descriptors can be classified into two types. The first type describes the features based on the 3D point cloud and its derivative data (such as *normal*, *curvature*) existing in a certain range going around a keypoint. As an example, Spin-images [10] establishes the tangent plane of every keypoint, and then projects the three-dimensional coordinates of a cylinder onto a two-dimensional spin image. Other descriptors describe the keypoint as a feature rather than an image by capturing the data from the descriptive range surrounding itself, such as Point Feature Histogram (PFH) [11] and Fast Point Feature Histogram (FPFH) [12]. They select large number of point pairs from  $k$  nearest neighbors of a keypoint and calculate parameters (such as *angle*, *normal*) to express the feature in the form of histogram. The second type is represented by Point Pair Feature (PPF) [13], [14] which requires to select two points (point pair) from the entire 3D model point set and couples them to define the 4D feature. Although there are plenty of hand-crafted 3D feature descriptors [15], these descriptors generally have poor performances in real applications due to their common restrictions, such as limited generalization power

in wide scenarios, inefficient computations and sensitivity to noise.

### B. DATA DRIVEN LEARNED 3D FEATURE DESCRIPTORS

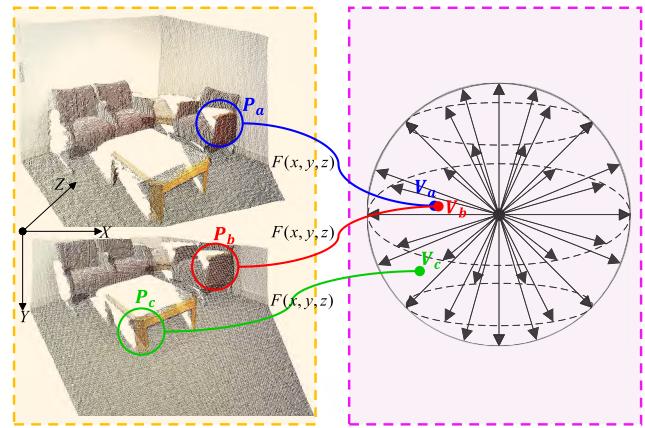
Many learning-based methods are proposed to extract 3D geometric descriptor along with the rise of deep learning. In terms of the data format: some methods [16], [17] try to extend depth information to CNN networks performing well in 2D images, whose processing data is RGB-D image. The multi-view images [18] have also been explored for 3D feature extraction and description. However, the data used in above-mentioned methods cannot represent 3D information directly. Some other methods try to directly operate on the 3D data. For example, [19], [20] are proposed to implement convolutions on the 3D volumetric data. [21] extends a novel operation called graph-convolution to process the mesh data. More recently, PointNet [5] and PointNet++ [22] are proposed to directly process raw point clouds. In terms of the application of these methods: some aim at 3D model classification [5], [23], [24], which pay more attention on the global feature of the 3D model. Some [25], [26] concentrate on the 3D semantic segmentation and part segmentation, which emphasize local feature descriptors and the representations of the relationship between point sets. The others get involved in diverse topics like 3D object detection and 6D pose estimation [27], [28], 3D model reconstruction and generation [29], and scene completion [30]. Many researchers try to generate more robust 3D feature descriptors based on data-driven methods. Unfortunately, there is still no successful descriptor that can generalize well in all scenarios.

### C. DEEP LEARNING FOR 2D/3D MATCHING TASKS

There are many researchers who attempt to leverage neural networks for matching tasks in the field of 2D images. As early as the nineties, Bromley *et al* [31] put forward the concept of “Siamese”, which is later extended by Y. LeCun to face verification task [32] and stereo matching task [33]. Inspired by MatchNet [34], Xiao *et al* firstly propose a novel network for learning local geometric descriptors in 3DMatch [8]. But the network only accepts TDF voxel grids data as input, which requires lots of memory. More recently, after the presence of PointNet [5], some works such as PPFNet [6], PPF-FoldNet [7] and 3DFeat-Net [35] are extended to directly process raw point clouds for 3D feature matching. However, these methods are still limited by the insufficiency of valid datasets and the absence of contextual structure information in disordered point cloud data, which result in low matching accuracy and poor generalization power in diverse scenarios.

### III. PROBLEM STATEMENT

The registration between two rigid point clouds refers to the task of estimating the transformation from the present point sets to the reference sets with 3D point correspondences. Given two 3D point clouds with overlapping parts:  $P = \{p_1, p_2, \dots, p_n\}$ ,  $Q = \{q_1, q_2, \dots, q_m\}$ ,  $P' \subseteq P$ ,  $Q' \subseteq Q$



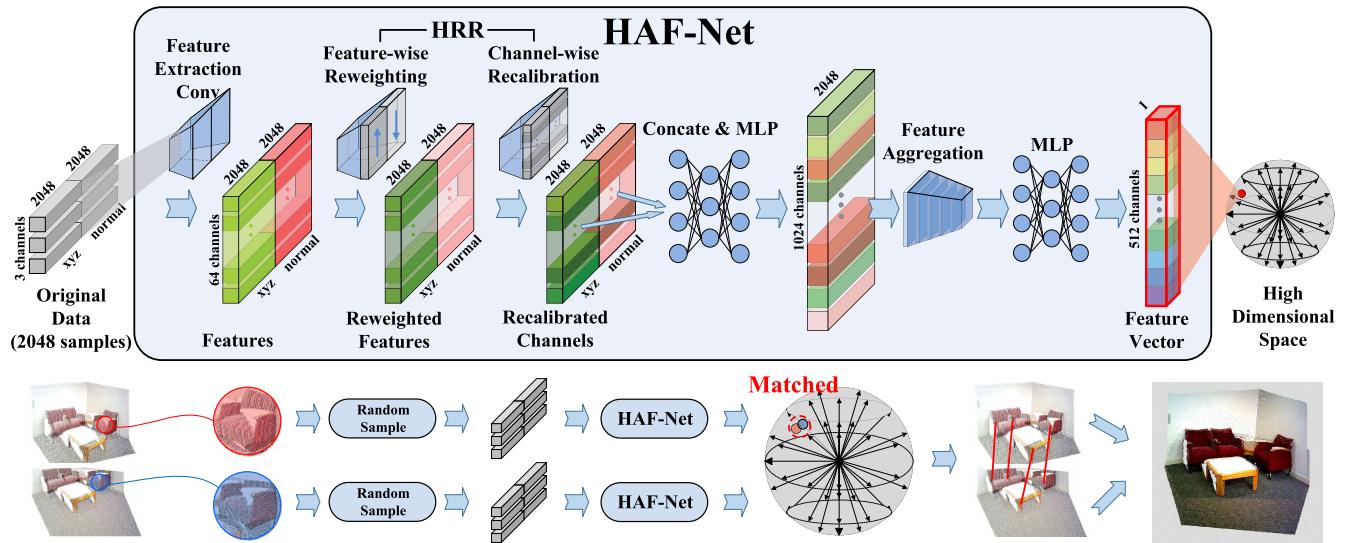
**FIGURE 2.** A schematic diagram of mapping point sets from low-dimensional space to high-dimensional feature space. The left region paled in orange represents the low-dimensional space. The right region paled in lilac indicates a high-dimensional feature space with  $n$  dimensions.  $F$  is the mapping function.  $P_a$  (blue),  $P_b$  (red),  $P_c$  (green) are three point sets from the low-dimensional space, and  $V_a$  (blue),  $V_b$  (red),  $V_c$  (green) are their mapped points represented by  $n$ -dimensional vectors in feature space.

and they are related by  $P' = R * Q' + t^\top + N_i$ , where  $R$  is a rotation matrix,  $t$  is a translation vector, and  $N_i$  is the noise vector. The objective is to find  $R$  and  $t$  by:

$$\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \| (p_i - (Rq_i + t^\top)) \|_2^2 \quad (1)$$

To obtain the optimal  $R$  and  $t$  which locally minimize the objective function, it is generally recommended to utilize ICP based algorithms. However, these methods are easily plunged into local minimums or even cannot converge due to improper initial values of  $R$  and  $t$ . If correspondence information between two point clouds can be directly obtained through the feature matching, we can easily compute the accurate closed-form solution of  $R$  and  $t$ . Therefore, the key component of the problem is to find a suitable 3D feature descriptor, which is utilized to obtain correct 3D correspondence between point sets. The essential difference between point cloud and image is that the former is unstructured. Thus it is hard to directly represent the geometric structure or other characteristics of local point sets.

The goal of the proposed method is to find a robust descriptor for the given point set. The problem can be formulated as finding a mapping function  $F_\theta$  which maps the point set from low-dimensional space to high-dimensional feature space. Assuming that  $P_a$  is a point set from a point cloud,  $P_b$  and  $P_c$  are the matched and unmatched point sets of  $P_a$  from another point cloud.  $V_a$ ,  $V_b$ ,  $V_c$  are their mapped points represented by  $n$ -dimensional vectors in feature space, where  $V_a = F_\theta(P_a)$ ,  $V_b = F_\theta(P_b)$ ,  $V_c = F_\theta(P_c)$ . There exist  $n$  points in each point set, and each point has low-dimensional information such as xyz. The  $F_\theta$  can be found by minimizing  $dis(V_a, V_b)$  and maximizing  $dis(V_a, V_c)$  simultaneously, where  $dis(\cdot)$  denotes



**FIGURE 3.** The architecture of our HAF-Net and the point cloud registration procedure with our HAF-Net. During training, the HAF-Net is applied on point block pairs with shared parameters.

the Euclidean distance. In other words,  $F_\theta$  should satisfy: 1)  $V_a$  is as close as possible to  $V_b$ ; 2)  $V_c$  is as far as possible from both  $V_a$  and  $V_b$ . The Fig. 2 shows an explicit instantiation.

#### IV. OUR HAF-NET

##### A. NETWORK ARCHITECTURE

Assuming that,  $P_{xyz}$  and  $P_{normal}$  represent the input of our model and the output is denoted by  $V_{out}$ . The entire process of our HAF-Net can be formulated as:

$$V_{out} = F_{haf}(P_{xyz}, P_{normal}) \quad (2)$$

where  $F_{haf}(\cdot)$  denotes our HAF-Net operation. Fig. 3 shows the architecture of our HAF-Net, consisting of three main components: 1) feature extraction module based on input transform and feature transform module from PointNet [5], 2) hierarchical feature reweighting and recalibration module (HRR) based on squeeze and excitation blocks for multi-level features fusion, 3) feature aggregation and compression structure based on NetVLAD [36] and fully connected layers.

The workflow of HAF-Net is as follows: 1) The features from multi-type information of the input data, such as position vector  $xyz$  and  $normal$ , are extracted by using the feature extraction structure. 2) Different types of features and different channel features contribute differently to the final descriptors. Through the proposed HRR module, the hierarchical attention of the multi-level features can be learned and fused at the same time. 3) After the fused feature maps are obtained, VLAD [9] is extended to aggregate and extract the significant features. Moreover, the features are compressed into a descriptive vector with several fully connected layers. Through above-mentioned steps, the representative descriptor of input point cloud block  $V_{out}$  is obtained.

##### 1) FEATURE EXTRACTION MODULE

The feature extraction module is based on the basic structure of PointNet [5], including input and feature transformations (T-net). Batchnorm layer is used for all layers with ReLU. The process can be modeled as:

$$T_{xyz} = F_{fe}(P_{xyz}), \quad T_{normal} = F_{fe}(P_{normal}) \quad (3)$$

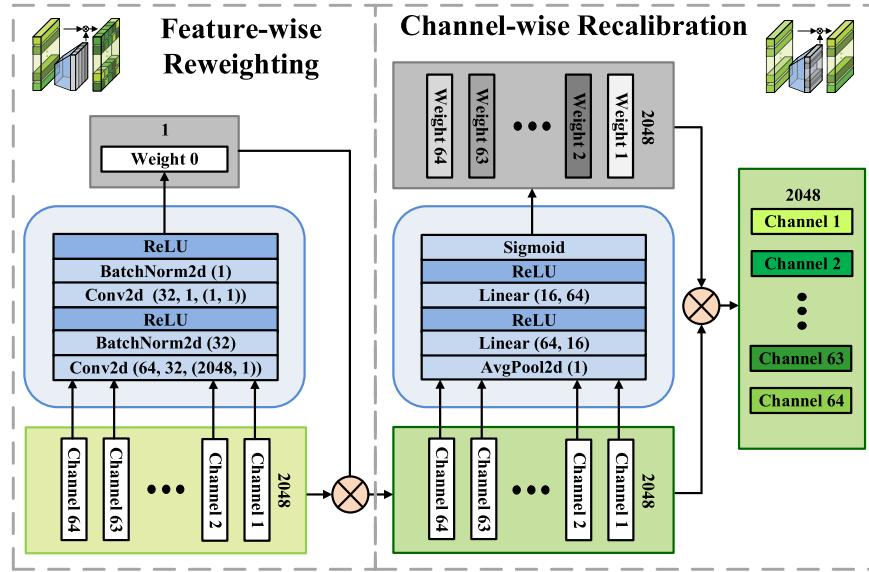
where  $F_{fe}(\cdot)$  denotes the operation of the feature extraction module.  $T_{xyz}$  and  $T_{normal}$  denote the output feature tensors of the feature extraction module from input  $P_{xyz}$ ,  $P_{normal}$  respectively.

##### 2) HRR MODULE: FEATURE-WISE REWEIGHTING

The proposed HRR module encompasses two parts, i.e., feature-wise reweighting and channel-wise recalibration. The first part, i.e., feature-wise reweighting block, is implemented to allocate different weights for multi-type information. This block can be developed as:

$$W_{fw} = F_{fw}(T_{xyz}, T_{normal}) \quad (4)$$

where  $F_{fw}(\cdot)$  represents the feature-wise reweighting operation. The  $W_{fw}$  is the output weight response of the  $T_{xyz}$  and  $T_{normal}$  from different types of inputs. To be more precise, the inputs of proposed network consist of the basic position information  $xyz$  and the auxiliary information such as  $normal$ . The inputs in different types might not contribute equally to expected feature descriptors. Therefore, to make the best use of multi-type information and lead to a stable convergence for the whole network, feature-wise reweighting block is proposed at hand to automatically learn suitable weights for inputs in different types. The feature-wise reweighting block is simple in structure but efficient (see details in Fig. 4). The low-dimensional input is firstly mapped into the high-dimensional feature space where representations in specific



**FIGURE 4.** The architecture of Hierarchical Reweighting and Recalibration Module (HRR) module with xyz as instance.

type can be obtained. Then the high-dimensional representations are remapped to a one-channel output, which indicates the new weight response of a specific input type. The learned weights are further utilized to multiply with the corresponding features to achieve the attention selection of different inputs. The feature-wise reweighting block uses the fully convolutional structure thus only a small number of parameters are added in terms of network complexity. Moreover, it supports backpropagation completely and can be implemented in all scenarios with multi-type inputs potentially.

### 3) HRR MODULE: CHANNEL-WISE RECALIBRATION

The second part, i.e., channel-wise recalibration, is implemented to learn to selectively emphasize features and suppress useless ones across channels. This block can be formulated as:

$$W_{cw\_xyz} = F_{cw}(T_{xyz}), \quad W_{cw\_normal} = F_{cw}(T_{normal}) \quad (5)$$

where  $F_{cw}(\cdot)$  represents the channel-wise recalibration block. The  $W_{cw\_xyz}$  and  $W_{cw\_normal}$  are the learned-channel activations of the  $T_{xyz}$  and  $T_{normal}$ . Specifically, we utilize Squeeze-and-Excitation (SE) [37] block to explicitly model the interdependencies between the channels of its convolutional features as shown in Fig. 4. The SE block takes the reweighted features as the input. Following the squeeze operation, the input is aggregated across spatial dimensions to produce a global channel descriptor, which embeds the global distribution of channel-wise feature response. This is followed by the excitation operation, where the squeezed descriptor is passed to a cascade of fully connected layers to generate new activations for each channel. In the end the learned-channel activations are used to recalibrate input

across channels achieving the attention selection of different channel-wise features.

The combination of feature-wise reweighting block and channel-wise recalibration block is declared to HRR module, which integrates the hierarchical attention of multi-level features. The complete HRR module can be summarized as:

$$T_{hrr} = W_{fw}(W_{cw\_xyz}T_{xyz} + W_{cw\_normal}T_{normal}) \quad (6)$$

where  $T_{hrr}$  is the output tensor of the HRR module. The “Hierarchical” not only indicates the different types of input data, but also means the diversity of feature channels. The low-dimensional input data can be transformed to representative features for subsequent layers through HRR module.

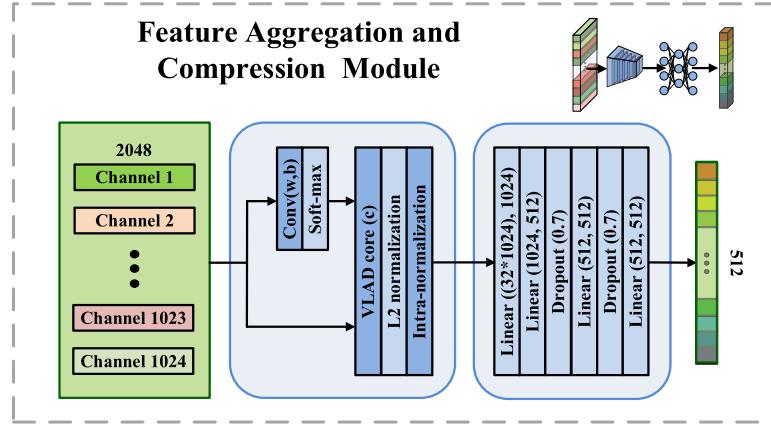
### 4) FEATURE AGGREGATION AND COMPRESSION MODULE

It is hasty to pool the max value of the feature map as the only primary feature. Our proposed method feeds the feature maps from the input point cloud into the NetVLAD [36] layers to aggregate the features. This feature aggregation structure is designed to aggregate  $N$  D-dimensional vectors into the  $(K \times D)$ -dimensional VLAD representation  $V$ , where  $D$  is the number of features channels. In order to learn the  $K$  cluster centers, denotes as  $\{c_k\}$ , we use basic convolution layers (shown in Fig. 5) to formulate the expression:

$$V(j, k) = \sum_{n=1}^N \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_k^T x_i + b_{k'}}} (x_i(j) - c_k(j)) \quad (7)$$

where  $\{w_k\}$ ,  $\{b_k\}$  are sets of trainable parameters for each cluster in  $\{c_k\}$ .

The paper extracts VLAD vectors via the feature aggregation structure from NetVLAD. The vector is a high dimensional vector for the input point cloud block. The higher dimension indicates more expensive computation complexity



**FIGURE 5.** The architecture of feature aggregation and compression module.

for nearest neighbor search. Therefore, we implement fully connected layers to compress the high-dimensional vector into a low-dimensional feature vector  $V_{out}$  to describe the point cloud block. The feature aggregation and compression module extracts principal components rather than the average value or the maximum of features, which makes our descriptor have stronger representation ability.

### B. DATASET PREPARATION

There is no standard datasets for 3D points matching except for 3DMatch [8] dataset. The 3DMatch dataset is in TDF format not point cloud. Moreover, many matched local point cloud pairs from 3DMatch test data are not really matched in the point cloud model. Namely, samples in 3DMatch dataset are in low quality and not suitable for training our network. Therefore, we take the insight of dataset generation from 3DMatch to produce our own dataset. In addition, we add two constraints to promise the correctness of matches, which further leads to better training data.

Inspired by 3DMatch [8], we generate the training data from the 3D reconstruction datasets with RGB-D images. The input to HAF-Net is a pair of point cloud blocks which are sampled from point cloud fragments. The ground truth of the input pair is the label indicating whether the blocks are matched or unmatched (“1” indicates matched and “0” indicates unmatched). To obtain training samples, a generic and efficient process is extended as follows: Firstly, point cloud fragments of every scene need to be registered based on the ground truth pose provided in the RGB-D datasets. Some points are randomly extracted from each fragments as interest points. Secondly, all existing fragments should be converted to a unified global coordinate system through their global poses. After that, the kd-tree algorithm is implemented to search correspondences in all fragments according to every interest point. To ensure that the views between observation pairs have a sufficient wide-baseline, each fragment are registered from 50 frames in average. Finally, the points around each interest point and their correspondences from

different fragments are extracted, which are further formed as matched pairs with positive label “1”. The unmatched pairs are randomly picked from any fragments as long as the distance between the two points is larger than 0.3 m.

In detail, each point cloud block has 2048 points which are equally sampled from a “ball sphere” with the interest point as center and the radius to be 0.6 m. The resolution of the fragments is 2cm and the fragments generated from RGB-D data which are surfaces rather than entities, the “ball sphere” theoretically has about 2800 points. Two constraints are given to eliminate the noise points and isolated points. One is that the blocks must be discarded if the number of points in the “ball sphere” is less than 2500. The other is that the matched pairs will be discarded if the difference between the numbers of points in their own “ball sphere” is more than 300. The correctness and reliability of the training data can be guaranteed with the above-mentioned constraints. Besides, the drift and noise from the depth sensor and the reconstruction datasets may cause week side-effects for the training, while improving the network generalization ability in some way.

The proposed data production procedure can help generate numerous data samples based on existing reconstruction datasets for the 3D points matching network. Similar to the dataset illustrated in 3DMatch [8], 54 different scenes collected from SUN3D [1], 7-Scenes [38], RGB-D Scenes v.2 [39], and BundleFusion [40] with totally over 200K RGB-D images are used. 46 scenes are used for training and 8 scenes for testing. There are about 160,000 block pairs in the training dataset with a 1:1 ratio between matches and unmatches. Each point of the block has 10-dimensional information including  $xyz$ ,  $rgb$ ,  $normal$ , and  $curvature$ , which is extracted from the fragment.

### C. IMPLEMENTATION DETAILS

#### 1) LOSS FUNCTION

Our HAF-Net is a Siamese architecture which takes two point cloud samples in same size as inputs and outputs a label

indicating the inputs are matched or unmatched. The loss function used during training is the contrastive loss [41] commonly used in Siamese networks. It can be formulated as:

$$L_{oss} = \frac{1}{2N} \sum_{i=1}^N yd^2 + (1-y) \max(\text{margin} - d, 0)^2 \quad (8)$$

where  $d$  is the Euclidean distance of two output descriptor vectors from the Siamese net, and  $y$  is the label. Standard contrastive loss uses the  $L_2$  norm. The disadvantage of  $L_2$  norm is that the outliers will be the main component of the loss in some situations. Therefore, we choose the piecewise loss function Huber Loss [42] to improve it. When  $L_{oss} \leq \delta^2$ , the loss is contrastive loss itself. Otherwise, the loss becomes a linear function about  $L_1$  norm like:

$$L_\delta = \frac{\delta}{N} \sum_{i=1}^N y |d| + (1-y) |\max(\text{margin} - d, 0)| - \frac{\delta^2}{2} \quad (9)$$

where  $\text{margin} = 2$  and  $\delta = 1$  during our training.

## 2) PRE-TRAINING AND TRAINING

The feature extraction module of our HAF-Net is partially based on PointNet, which is originally utilized for 3D object classification and segmentation. In experiments, the HAF-Net network with many BN (Batch Normalization) layers cannot always converge stably in the proposed dataset. We analyse that it is hard to direct teach a deep network to recognize which pairs are matched or not. In other words, currently there are no fundamental pre-trained networks which can extract generic 3D features of point clouds like what VGG [43] and ResNet [44] do in 2D images. The network should learn some basic 3D features first. In order to make the network converge stably and rapidly, we pre-trained our network on 3D object classification datasets analogizing the way VGG/ResNet used.

Firstly, we amassed a 3D classification dataset using the method similar to the data production process introduced in Sec. IV-B. In fact, in Sec. IV-B, every sampled point cloud block has more than one matches from many different views. By manual selection and data enhancement, we selected some basic 3D shapes, such as planes, wall corners, table corners and so on, from these blocks and used them as classes of samples to form the final 3D object classification dataset. The dataset consists of 159 classes, and each class has about 500 samples where only one tenth of them are original samples from fragments and the rest samples are from data enhancement. The process of the sample enhancement is as follows: 1) adding Gaussian noise to some points of the point cloud block, 2) translating or rotating the whole block randomly, 3) disordering the points of the block randomly. Secondly, our HAF-Net for classification was trained over the 3D classification dataset amassed before by minimizing the NLL (Negative Log Likelihood) loss. After the training of classification network converged, the parameters of the

pre-training model were used to initialize common layers of proposed HAF-Net.

By adding the pre-training procedure, the training process of HAF-Net converged steadily. Specifically, the proposed HAF-Net was implemented based on PyTorch and trained on one NVIDIA Titan-X GPU using the Adam optimizer with a batch size of 32 and a learning rate of  $10e^{-4}$ . The training process took about 3 hours for 4 epochs on pre-training and 10 hours for 10 epochs on training.

## V. EVALUATION

In Sec. V-A, we evaluate the ability of our local descriptors on keypoint matching and conduct a series of ablation studies to verify the capabilities of the two key modules in our HAF-Net. To validate the performance of our descriptor, we apply the descriptor on point cloud registration for real scene reconstruction, in Sec. V-B. In Sec. V-C additional experiments verify the generalization ability of our network in 3D point cloud classification.

### A. KEYPOINT MATCHING

We follow the procedure in [8] to evaluate the quality of our 3D descriptor on keypoint matching. We construct a corresponding test dataset with the methodology presented in Sec. IV-B. The test dataset contains totally 20,000 local point cloud pairs with a 1:1 ratio between matches and unmatches. These pairs are from the same 8 scenes in the 3DMatch benchmark [8].

#### 1) ADVANTAGES OF OUR DESCRIPTOR

The comparison between our descriptor and several state-of-the-art 3D descriptors are compared with ours on 3DMatch benchmark [8]. One is the most popular traditional hand-crafted 3D descriptor FPFH [12]. We use its standard implementation provided in the Point Cloud Library (PCL). Another is the 3DMatch [8]. We use the codes they provide to generate our point cloud pairs into TDF and get their descriptors. PointNet [5], as our baseline network, is also used for contrast.

**TABLE 1.** Keypoint matching error (%) at 95% recall.

Method	Error(%)	XYZ	Normal
FPFH [12]	61.3	yes	yes
3DMatch [8]	27.3	yes	no
PointNet [5]	22.3	yes	no
HAF-Net	<b>16.3</b>	yes	no
HAF-Net (with normal)	<b>13.2</b>	yes	yes

We use the false-positive rate (error) at 95% recall as the evaluation metric in Tab. 1 (the lower the better). Experimental results show that our descriptor performs significantly better than the handcrafted descriptor FPFH. Our descriptor expectedly outperforms our baseline PointNet by 6% and 3DMatch by more than 10%, which proves the effectiveness of our designed HAF-Net. In addition, our descriptor learned from both *xyz* and *normal* yields a lower error rate than

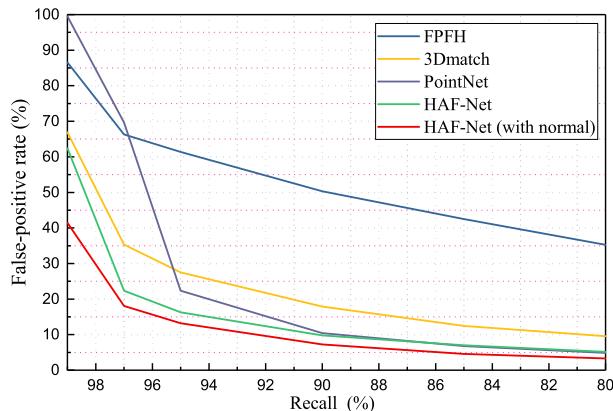


FIGURE 6. Keypoint matching error (%) at different recalls.

all other descriptors. It further illustrates that our HAF-Net can adaptively fuse the multi-type information from different input data based on hierarchical attention mechanism. Fig. 6 shows the false-positive rate at different recalls, which proves that our descriptor performs better than others.

## 2) EFFECTIVENESS OF DIFFERENT COMPONENTS

Further experiments are carried out to explore the contribution of two key components in our HAF-Net: 1) HRR module, 2) Feature Aggregation and Compression module (“FAC” for short to make the following discussion convenient). We still apply the false-positive rate at 95% recall as evaluation metric to keep the consistency of the experiments.

TABLE 2. Effect of different components in performance: values depict the false-positive rate (error) at 95% recall on 3DMatch benchmark [8]. “Base” refers to our baseline PointNet [5]. “FAC” refers to the feature aggregation and compression module. “SE” and “HRR” refer to the squeeze-and-excitation block [37] and our HRR module.

	xyz	xyz & normal
Base	21.6%	20.2%
Base + FAC	20.8%	19.4%
Base + SE	17.1%	16.5%
Base + HRR	-	<b>14.9%</b>
Base + SE + FAC	16.3%	15.9%
Base + HRR + FAC	-	<b>13.2%</b>

Results are revealed in Tab. 2, containing three columns which respectively indicate the model and quantitative performances of different inputs from left to right. In the middle column, only *xyz* is used for training, and our baseline (Base) is Pointnet. We separately add proposed FAC and SE to the baseline. The improvement of SE is larger than the improvement of FAC (4.5% V.S. 0.8%). Then, we add both FAC and SE to the baseline to validate the improvement of the complete setting. The model with both two modules has a better performance than any single module. In the right column, both *xyz* and *normal* are used for training. Based on the experiments we implemented for *xyz*, we further add proposed HRR module to the baseline. Compared with SE module, HRR module achieves more improvements.

We visualize the attention weights from the feature-wise reweighting block and find that the weight of *xyz* is higher than that of *normal*. Though simple in structure, the HRR module can make better fusion of multi-level attentions, especially where the input data contains different types of information.

Based on the ablation studies above, the combination of our two components (HRR and FAC) achieves outstanding results by fusing the attention of hierarchical features and compressing the aggregated descriptor.

## B. POINT CLOUD FRAGMENTS REGISTRATION

Considering that our ultimate goal is to increase the accuracy of point cloud registration, we evaluate the practical use of our descriptor in point cloud fragments registration. Due to the difference of the point-cloud resolution, the randomness of RANSAC and the numerical instability of the transformation solver, unknown and non-repeatable errors will be introduced to the final transformation matrices. Sharing the same idea with [6], we think it is unfair to use the estimation accuracy of rigid transformation matrices to evaluate the function of descriptors. Because the descriptors are only used in the matching stage of point cloud registration. We utilize the same evaluation scheme introduced by Deng *et al* [6]. The precision can always be improved by better corresponding pruning if enough right matches can be found. Therefore, we directly compute the recall by averaging the number of matched fragments across datasets in the benchmark. The specific formula is defined as follow:

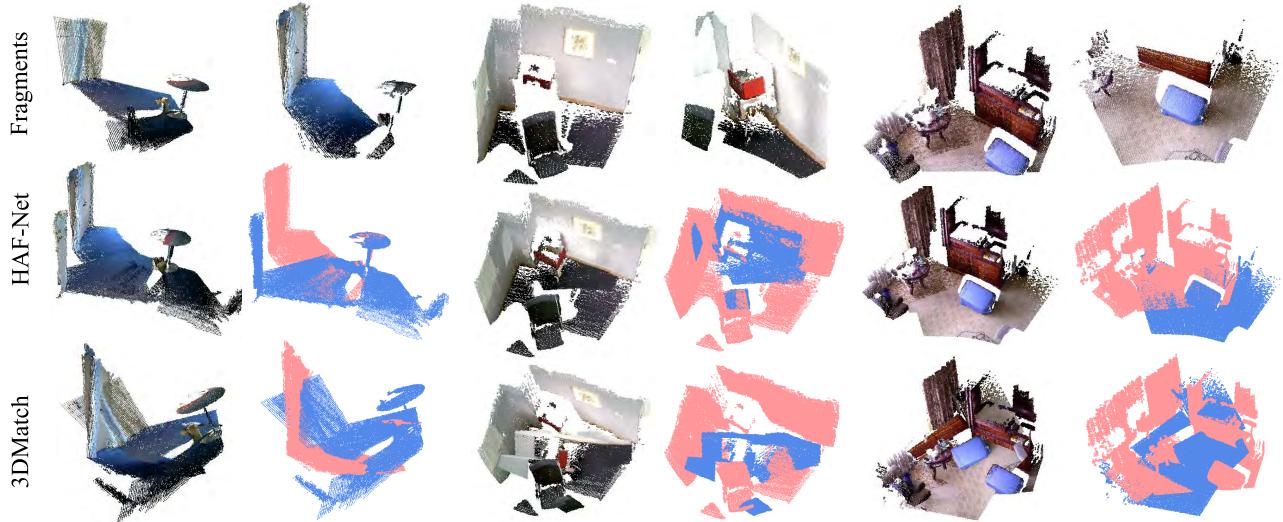
$$R = \frac{1}{M} \sum_{k=1}^M \left( \left[ \frac{1}{M_\Omega} \sum_{i,j=1}^{M_\Omega} (\| T_{ij} p_i - q_j \|^2 < \tau_1) \right] > \tau_2 \right) \quad (10)$$

where  $M$  is the number of ground truth matched fragment pairs which have at least 30% keypoints overlap with each other under ground-truth transformation  $T$  within distance  $\tau_1 = 10cm$ .  $i, j$  denote elements of the found correspondence set  $\Omega$ .  $M_\Omega$  is the total number of found matches in  $\Omega$ .  $p$  and  $q$  respectively come from the first and second fragment under matching. The inlier ratio is set as  $\tau_2 = 0.05$ .

We have evaluated our method against the hand-crafted descriptor FPFH [12], the vanilla PointNet [5], as well as the state of the art deep learning based 3D feature descriptor 3DMatch [8] on the reproduced evaluation experiment according to the procedure in [7]. As seen from Tab. 3, our descriptor works best in five of the eight scenes for 3D point matching. Our descriptor drastically outperforms the hand crafted FPFH on the average recall. There are 3.8% improvement on the average recall over the 3DMatch. Our method expectedly outperforms our baseline PointNet by 10.8%. Due to the source codes of PPFNet [6] and PPF-FoldNet [7] have not been released, we put the results of PPFNet and PPF-FoldNet (using 2K sample points) from paper [7] into Tab. 3, which are marked with \*. Compared with their results, our HAF-Net is clearly superior.

**TABLE 3.** Evaluations of fragments registration on benchmark from 3Dmatch. kitchen is from 7-scenes [38] and the rest from SUN3D [1].

Method	Kitchen	Home1	Home2	Hotell1	Hotel2	Hotel3	Study room	MIT lab	Average
FPFH [12]	41.2%	62.3%	51.1%	60.7%	48.3%	42.9%	45.0%	51.3%	50.4%
3DMatch [8]	60.8%	77.8%	65.3%	64.6%	<b>66.8%</b>	71.4%	58.7%	62.9%	66.0%
PointNet [5]	50.4%	70.5%	74.6%	68.0%	55.2%	43.5%	46.5%	63.6%	59.0%
PPFNet [6]*	<b>89.7%</b>	55.8%	59.1%	58.0%	58.0%	61.1%	53.4%	63.6%	62.3%
PPF-FoldNet(2K) [7]*	73.5%	75.6%	62.5%	65.9%	60.6%	<b>88.9%</b>	57.5%	59.7%	68.0%
HAF-Net	58.4%	74.1%	76.7%	73.3%	60.4%	46.2%	<b>70.5%</b>	<b>86.3%</b>	68.2%
HAF-Net(with normal)	58.9%	<b>80.3%</b>	<b>80.9%</b>	<b>73.3%</b>	58.6%	57.1%	67.6%	81.3%	<b>69.8%</b>

**FIGURE 7.** Visualization of point cloud fragments registration. Fragments (top row) have drastic viewpoint differences which is challenging for registration. 3DMatch [8] fails at aligning the fragments. While HAF-Net (middle row) is able to successfully align each pair of fragments by matching its robust descriptors.

Moreover, there are some possible factors that can make our reproduced evaluation experiments and the evaluation experiments in [7] incompletely equivalent: 1) the method used to obtain initial correspondence set; 2) the number of interest points extracted from each point cloud fragment (we followed the principle in 3DMatch to extract 500 points from each point cloud fragment as interest points); 3) the maximum iterations of RANSAC. [7] sets the maximum iterations of RANSAC to be 50000, while only 1000 iterations are used in our paper. In this case, the performance of our descriptor is still competitive.

To further prove the benefits of our descriptors on point cloud registration, we calculate the transformation between fragments based on the matching pairs achieved above, using the matching pipeline with RANSAC (1000 iterations and less than  $0.06m$  as interior-point). The registration is considered correct when the RTE (Relative Translational Error) and RRE (Relative Rotation Error) [45] are both below a predefined threshold of  $0.05m$  and  $1^\circ$ . Our experimental results attained a 59.2% accuracy for fragments registration at the 69.8% recall on average. Fig. 7 shows some of the fragments registration results, which proves that our descriptor performs stable for the point cloud registration even in some challenging scenarios. Furthermore, there are example results of the

fragment registration in three scenarios with varying degrees of viewpoint differences shown in Fig. 8: (a) One fragment is completely overlapped by the other; (b) Two fragments overlap each other partially; (c) Two fragments have a few overlaps.

### C. GENERALIZATION CAPABILITY VERIFICATION IN 3D OBJECT CLASSIFICATION

In this experiment, the generalization ability of our descriptor is evaluated. We directly applied our network to the 3D object classification task on the ModelNet40 [46] dataset.

*ModelNet40:* There are 12,311 CAD models of 40 categories (mostly man-made). We use the official split with 9,843 shapes for training and 2,468 for testing, similar to [22].

Specifically, the loss function, optimizer, and the learning rate are completely retained from the official source code given by PointNet [5]. All the training and testing parameters involved are same, such as the batch size and the number of epochs. We train our net without any parameters adjusting and optimizing on the ModelNet40 dataset. The results are shown in Tab. 4. Our HAF-Net achieves a better performance than both the vanilla PointNet and the hierarchical PointNet. The HAF-Net with both the *xyz* and the *normal*



**FIGURE 8.** Example results of the fragment registration in three scenarios with varying degrees of viewpoint differences. (a) One fragment is completely overlapped by the other. (b) Two fragments overlap each other partially. (c) Two fragments have a few overlaps.

**TABLE 4.** 3D shape model classification results on modelNet40.

Method	Accuracy(%)
PointNet (vanilla) [5]	87.2
PointNet [5]	88.8
PointNet++ [22]	90.7
PointNet++ (with normal) [22]	91.9
HAF-Net	89.2
HAF-Net (with normal)	90.7

inputs can easily achieve the similar accuracy as PointNet++ even though our network added no multi-scale and multi-resolution grouping structures. The results above show that our network has a good generalization ability on new tasks and different datasets.

## VI. CONCLUSION

In this work, we present the HAF-Net to generate a hierarchical attention fused 3D descriptor for 3D point matching. A global convolutional kernel is implemented to transform feature maps (from one data type such as xyz) to 1D response values. Then the values in multiple channels are fused to a single value through a convolutional layer. The single value is later multiplied with corresponding feature maps. Our reweighting module can automatically learn a response value to weight the importance of each data type and fuse the different data properly and efficiently. Experimental results illustrate that our learned descriptor achieves state-of-the-art performance on 3D point matching and point cloud fragments registration. The HRR module proposed for hierarchical attention fusion can adaptively integrate multi-level features through learning. This module can be further implemented in other applications, such as the fusion of multi-modal information and multi-task. The combination of VLAD and fully connected layers aggregates and compresses fused features into a low dimensional descriptor, which has a stronger representation ability. Moreover, the proposed dataset might potentially facilitate future research that is involved in 3D feature representation and 3D point matching.

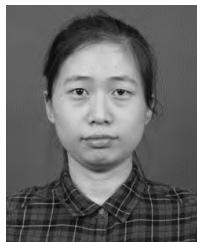
## ACKNOWLEDGMENT

Project supported by Shanghai Municipal Science and Technology Major Project (Grant No. 2018SHZDZX01, ZHANGJIANG LAB).

## REFERENCES

- [1] J. Xiao, A. Owens, and A. Torralba, “SUN3D: A database of big spaces reconstructed using SfM and object labels,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1625–1632.
- [2] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [3] P. C. Ng and S. Henikoff, “SIFT: Predicting amino acid changes that affect protein function,” *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3812–3814, 2003.
- [4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
- [5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.
- [6] H. Deng, T. Birdal, and S. Ilic, “PPFNet: Global context aware local features for robust 3D point matching,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 195–205.
- [7] H. Deng, T. Birdal, and S. Ilic, “PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2018, pp. 620–638.
- [8] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, “3DMatch: Learning local geometric descriptors from RGB-D reconstructions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 199–208.
- [9] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.
- [10] A. E. Johnson, “Spin-images: A representation for 3-D surface matching,” Carnegie Mellon Univ., Pittsburgh, PA, UAS, Tech. Rep. CMU-RI-TR-97-47, 1997.
- [11] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, “Aligning point cloud views using persistent feature histograms,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3384–3391.
- [12] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2009, pp. 3212–3217.
- [13] B. Drost and S. Ilic, “3D object detection and localization using multi-modal point pair features,” in *Proc. 2nd Int. Conf. 3D Imag., Modeling, Process., Vis. Transmiss.*, Oct. 2012, pp. 9–16.
- [14] T. Birdal and S. Ilic, “Point pair features based object detection and pose estimation revisited,” in *Proc. Int. Conf. 3D Vis.*, Oct. 2015, pp. 527–535.
- [15] M. Khouri, Q.-Y. Zhou, and V. Koltun, “Learning compact geometric features,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 153–161.
- [16] M. Schwarz, H. Schulz, and S. Behnke, “RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features,” in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2015, pp. 1329–1335.
- [17] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from RGB-D images for object detection and segmentation,” in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 345–360.
- [18] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3D shape recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 945–953.
- [19] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, “FPNN: Field probing neural networks for 3D data,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 307–315.
- [20] B. Graham and L. van der Maaten, “Submanifold sparse convolutional networks,” Jun. 2017, *arXiv:1706.01307*. [Online]. Available: <https://arxiv.org/abs/1706.01307>
- [21] L. Yi, H. Su, X. Guo, and L. Guibas, “SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation,” in *Proc. CVPR*, Jul. 2017, pp. 6584–6592.
- [22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
- [23] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng, “Convolutional-recursive deep learning for 3D object classification,” in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2012, pp. 656–664.
- [24] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “Shapenet: An information-rich 3D model repository,” Dec. 2015, *arXiv:1512.03012*. [Online]. Available: <https://arxiv.org/abs/1512.03012>
- [25] L. Yi *et al.*, “Large-scale 3D shape reconstruction and segmentation from shapenet core55,” Oct. 2017, *arXiv:1710.06104*. [Online]. Available: <https://arxiv.org/abs/1710.06104>
- [26] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, “3D recurrent neural networks with context fusion for point cloud semantic segmentation,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 415–430.
- [27] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, “Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 205–220.
- [28] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, “Multi-view self-supervised deep learning for 6D pose estimation in the Amazon picking challenge,” in *Proc. IEEE Int. Conf. Robot. Automat.*, May/Jun. 2017, pp. 1383–1386.

- [29] P. Achlioptas, O. Diamanti, I. Mithiagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," Jun. 2018, *arXiv:1707.02392*. [Online]. Available: <https://arxiv.org/abs/1707.02392>
- [30] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner, "ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans," in *Proc. CVPR*, vol. 1, Jun. 2018, pp. 4578–4587.
- [31] J. Bromley, I. Guyon, Y. LeCun, and E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," in *Proc. 6th Int. Conf. Neural Inf. Process. Syst.*, Nov./Dec. 1993, pp. 737–744.
- [32] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2005, pp. 539–546.
- [33] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, nos. 1–32, p. 2, Jan. 2016.
- [34] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3279–3286.
- [35] Z. J. Yew and G. H. Lee, "3DFeat-Net: Weakly supervised local 3D features for point cloud registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2018, pp. 630–646.
- [36] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. 29th IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5297–5307.
- [37] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," Sep. 2017, *arXiv:1709.01507*. [Online]. Available: <https://arxiv.org/abs/1709.01507>
- [38] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2930–2937.
- [39] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3D scene labeling," in *Proc. IEEE Int. Conf. Robot. Automat.*, May/Jun. 2014, pp. 3050–3057.
- [40] A. Dai, M. Nießner, and M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundle-Fusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 4, p. 76a, Jul. 2017.
- [41] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 1735–1742.
- [42] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.
- [43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [45] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
- [46] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.



**DONGCHEN ZHU** received the Ph.D. degree from the Shanghai Institute of Microsystem and Information Technology, University of Chinese Academy of Sciences, China, in 2018. She is currently an Assistant Professor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, China. Her current research includes computer vision, stereo vision, 3D reconstruction, and artificial intelligence.



**LIANG DU** received the B.S. degree from Harbin Engineering University, China, in 2016. He is currently pursuing the master's degree in information and communication engineering with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences; and the University of Chinese Academy of Sciences. His current research interests include monocular depth estimation and object detection.



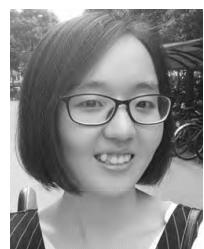
**GUANGHUI ZHANG** received the B.S. degree from the South China University of Technology, China, in 2016. She is currently pursuing the Ph.D. degree in information and communication engineering with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences; and the University of Chinese Academy of Sciences. Her current research includes stereo vision, 3D reconstruction, and 3D scene understanding.



**JIAMAO LI** received the Ph.D. degree from the Tokyo Institute of Technology, Japan, in 2012. He is currently a Professor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences; and the University of Chinese Academy of Sciences. His current research interests include computer vision, machine vision, 3D micro-imaging, and artificial intelligence.



**XIAOLIN ZHANG** received the Ph.D. degree from Yokohama National University, in 1995. He was a Professor with the Tokyo Institute of Technology, Japan, from 2012 to 2013. He is currently a Professor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences; and the University of Chinese Academy of Sciences. He is also a Distinguished Adjunct Professor with the School of Information and Technology, ShanghaiTech University. His research interests include bionics, brain science, computer vision, and artificial intelligence.



**WENJUN SHI** received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, China, in 2015. She is currently pursuing the Ph.D. degree in information and communication engineering with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences; and the University of Chinese Academy of Sciences. Her current research interests include 3D reconstruction, 3D scene understanding, and visual odometry.