Reference Grid-assisted Network for 3D Point Signature Learning from Point Clouds

Jing Zhu Yi Fang*
NYU Multimedia and Visual Computing Lab, USA
New York University, USA
New York University Abu Dhabi, UAE
{jingzhu, yfang}@nyu.edu

Abstract

Learning a robust 3D point signature from point clouds is an interesting but challenging task in the computer vision field due to the irregular and unordered structure characteristics of the point cloud data. In this paper, we propose to learn a 3D point signature by exploring the implicit relation between keypoints and their neighbors (grouped as patches) among the given scene point clouds. We design a uniform reference grid to represent the raw relation between each keypoint and its neighbors from the raw point clouds. In order to learn a 3D point signature gradually from expanding perceptive region, we create a novel siamese framework with a multi-layer perceptron (MLP)-based unit feature network and a 3D convolutional neural network (CNN)-based grid feature network. Specifically, the unit feature network aims to dig the connections among points fallen into the same unit of the reference grid, while the grid feature network is used to discover the grid-wise relations across the whole reference grid with concatenation of the learned unit-wise features. Moreover, we introduce an attention network upon the unit feature network to enhance the discriminative ability of our learned 3D point signature. Our proposed 3D point signature achieves superior performance over other state-of-the-art methods on keypoint matching and geometric registration on the real-world scenes datasets, e.g. SUN3D, 7-scenes and the synthetic scan augmented scenes in ICL-NUIM dataset. More importantly, our learned 3D point signature successfully handles the point cloud fragment alignment challenges by producing correct transformations with RANSAC algorithm.

1. Introduction

A robust 3D point signature has great effects on a variety of applications in the 3D computer vision field, especially for typical local matching problems, such as keypoint matching, scene geometry registration and mesh registration. In the early attempts, researchers focus on designing hand-crafted features solely based on the geometry structure of the 3D mesh. In recent years, learning a 3D point signature using deep learning techniques attracts great attention from the computer vision community, especially with the advances of convolutional neural network (CNN) techniques. Given the nature that CNN is designed for 2D, most researchers convert the 3D data (e.g. mesh, point clouds) into 2D format representations for better signature learning, such as multi-view rendered images or hand-crafted geometry-based features. The development of 3D CNN enables researchers to learn a 3D point signature directly from a 3D format (voxelized) data. 3DMatch [33] is the first work to learn a 3D local signature from voxelized data using 3D

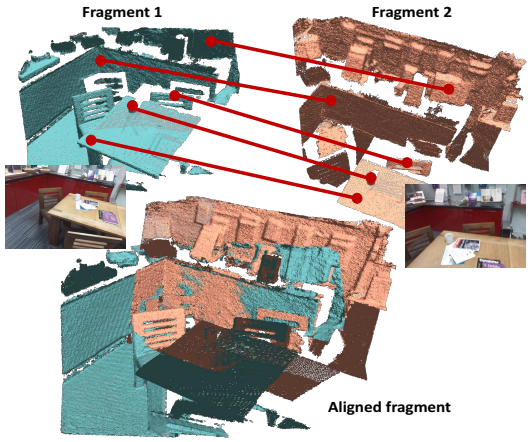

Figure 1: In our work, we learn a robust 3D point signature from raw scene point clouds, which can be used to recognize the match keypoints from same part of a scene but in different fragments. Then, the point cloud fragments can be further aligned by the estimated transformation computed from the match keypoint pairs.

*indicates corresponding author.

CNN, where they sampled keypoints and extracted voxlized local patches from depth images. Though their work is impressive, it requires large amount of computation time when preparing and processing the voxelized data. The high memory consumption also limits the resolution of the voxelized patch, which disadvantages the 3D local signature learning.

Recently, the success of PointNet [22] provides us a hint to learn a robust 3D point signature from more natural 3D point cloud data using a MultiLayer Perceptron(MLP)-based network. PPFNet [8] is the first work to utilize the PointNet technique on 3D local signature learning for the real-world scene geometry registration task. In addition to the 3D point (XYZ) coordinate values, they also feed the computed normals and hand-crafted point pair features (PPF) together into their model to learn a point signature. Their latter work PPF-FoldNet [7] obtains a even better performance only using PPF as their network inputs, implying that the hand-crafted PPF features actually contribute much more than the 3D point (XYZ) coordinates in their point signature learning. Therefore, it still remains a very challenging problem how to learn a robust 3D point signature from the raw point (XYZ) coordinates without any auxiliary hand-crafted features.

In this paper, we present a model that learns a discriminative 3D point signature from point (XYZ) coordinates, without any precomputing point features or normals. Taking advantages of the relation of keypoints and their neighbor points, we first group the keypoints and their neighbors within certain radius as patches to have a larger view area, then we applied a reference grid on the patches. For each unit in the reference grid, we compute the coordinate differences between the center of the unit and its N-nearest neighbor points from the patches as unit values. Later, the reference grid with coordinate differences is fed into a MLP-based network to learn unit-wise features. We concatenate the unit-wise features with the original reference grid, and introduce a 3D CNN-base network to learn a grid-wise feature as the final 3D point signature. Therefore, our 3D point signature can be gradually learned from a smaller unit region to a larger grid view. In addition, an attention network is added on the MLP-based unit feature network to strengthen the discriminative ability of the point feature. In order to better learn a 3D point signature for matching problems, we develop each network component in a siamese manner, and train all the networks simultaneously with a contrastive loss.

To validate our proposed method, we conduct experiments on the 3DMatch dataset [33] for two 3D matching tasks, i.e., keypoint matching and geometric registration. We provide the quantitative results of keypoint matching task on 3DMatch testing dataset. For the geometric scene fragment registration task, we report the quantitative comparisons on both real-world scan scenes and synthetic augmented scenes. Moreover, we visualize some fragment alignment results for qualitative comparison. The experimental results demonstrate that our proposed method successfully learns a discriminative 3D local signatures from point clouds (XYZ coordinates) with superior performance over other state-of-the-art methods. We also discuss the effectiveness of the introduced attention network by providing the quantitative results when training our feature learning network without the attention part.

In summary, the main contributions of our work are concluded as:

- To address the challenging 3D matching problems, we propose to build a novel end-to-end siamese-framework to learn a robust 3D point signature from point clouds (XYZ coordiantes) without any precomputed hand-crafted point features as auxiliary.
- Specifically, we design a reference grid to encode the relation between the keypoints and their neighbor points by computing the coordinate differences between the unit centers and its n-nearest points.
- We develop a MLP-based unit feature network followed by a 3D CNN-based grid feature network to learn a robust 3D point signature from multiple perceptive regions. An attention network is introduced to enhance the discriminative ability of our 3D point signature.
- The experimental results demonstrate that our proposed framework can generate robust discriminative 3D point signature and achieves superior performance over the state-of-the-art methods on both keypoint matching and geometry registration.

2. Related Work

To find a local signature that can well describe a given keypoint, researchers started their works on designing a hand-crafted signature based on the geometric structure of a 3D mesh. In the well-known spin image [17] method, Johnson et al. proposed a signature that consisted of a set of 3D points, surface normals and spin images generated from the oriented points of 3D meshes. Statistic analysis [2, 34, 10] was another popular direction to define local signatures for points of the given 3D objects for shape registration. For example, Zhang et al. [34] got their signatures by computing the distributions of the average geodesic distances between points over the meshes due to their observation that geodesic distance worked well on the problems dealing with graph-like structures. Besides the geodesic distances distribution, there were many other hand-crafted histogram-based signatures computing on the curvatures, diameters, and other geometrical properties [10, 28, 26, 25, 24].

Though the above hand-crafted signatures were inspired, they were all designed on the 3D mesh data with clear structure that is composed of 3D points and triangles connected by the points. However, there is no connection between points in the raw point clouds, which limits the direct utilization of the hand-crafted signatures on point cloud data. In our work, we aim to find a robust 3D point signature from the point cloud data with irregular structure.

In addition to the traditional hand-crafted signatures, how to obtain a robust learning-based signature has attracted a lot of attention from the computer vision community, due to the great success of applying deep learning techniques on various applications, such as retrieval, classification and transfer learning [19, 18, 27, 36, 11, 9, 14, 6, 5]. Since the classic convolutional neural network was developed for addressing 2D image-based tasks, it is difficult to directly apply a CNN network on 3D data. In some attempts [12, 32, 15, 3, 4, 20, 21], researchers preprocessed the 3D objects by converting those objects into some hand-crafted features, or rendered images (from different view angles), so that they can build a CNN network upon the extracted features or images.

The popularization of 3D CNN techniques enables the researchers to construct a deep convolutional network on more natural 3D voxel data for 3D point signature learning. 3DMatch [33] was the first work that learned a point signature from depth scans using a 3D CNN on volumetric point patches. Each voxel of the point patch contained the truncated distance between the voxel center and the nearest neighbor point. However, their matching performance was far from satisfactory. The causes could be that they 1) computed the patch values from the nearest neighbor leading to a very limited perspective area for each unit in the patch, and 2) ran a 3D CNN network on the extracted patches with 3D kernels to learn a patch-wise feature but ignored the influences of points fallen within each unit of the patch. Therefore, we are seeking a way to better learn a 3D point signature by taking both the influences of points within each unit and the connections of units among the whole patches into consideration. The effectiveness of learning features progressively from different receptive field has also been validated in VoxelNet[35] on 3D object detection task.

Beside the models working on 3D voxelized data, inspired by the successes of PointNet [22] and PointNet++ [23], researchers also turned their eyes on learning 3D point signatures from point cloud data. Most of them focused on digging the implicit relation between points and their neighbor points (e.g. normals, point pair features, coordinate differences, normal angle differences) or computing some extra point features along with the point (XYZ) coordinates to boost the performance mostly on segmentation tasks [30, 29, 16].

Specifically, Deng et al. paid their attention on the matching problems, and proposed PPFNet [8] to learn a 3D point signature from 3D point clouds. Apart from the point (XYZ) coordinates for keypoints and neighbor points, they also computed the normals and hand-crafted point-pair-features (PPF) as auxiliary. The PPF feature was critical to their matching performance, since it provided much more information (e.g. normal angle differences) than simple point (XYZ) coordinates differences. In their latter work PPF-FoldNet [7], they obtained a better performance even only using the PPF, which also proved the larger contribution of the hand-crafted PPF on their matching performance. Another recent work 3DFeat-Net[31] learned the point signatures solely in point-wise manner using MLP but ignored the structure information, and their two-stage model needed to take the point cloud of entire scenes as input, which is costly in both memory and time. In our work, we aim to build a one-stage model that works directly on the (XYZ) coordinates of the keypoint and neighbor points without any extra support, e.g. normal, color, hand-crafted features.

3. Approach

Considering that neighbor points can provide more information for given single keypoint, instead of learning a point signature from single point (XYZ) coordinates, we group all the neighbor points around the keypoint within a radius as patches (3D local neighberhood) to learn a point signature, analogous to 2D image patches around detected keypoints in 2D feature matching frameworks. Then, for each input patch, we extract a reference grid and build our networks upon the extracted reference grids to learn a robust 3D point signature from progressively expanding perceptive region.

As shown in Fig. 2, there are mainly four components in the framework, 1) reference grid extraction that generate the reference grids for any input patches, 2) a siamese unit feature network to explore the the implicit relation among neighbor points within each unit, 3) an attention network added upon the siamese unit feature network to improve the learning ability of the unit feature network, 4) a siamese grid feature network that finally learns a robust 3D signature over the whole grid after concatenating the generated unit-wise features with the original extracted grid. All the network components are connected by a contrastive loss defined on the final output of the grid feature network. The technical details for each component are provided separately below.

Reference Grid Extraction In order to better dig the influence of points within the patches, we decide to learn features gradually from smaller perceptive regions to larger ones using reference grids for convenience. More importantly, coordinate differences between the unit centers and points can provide us some hints on the patch density. The pipeline of the reference grid extraction part is depicted in

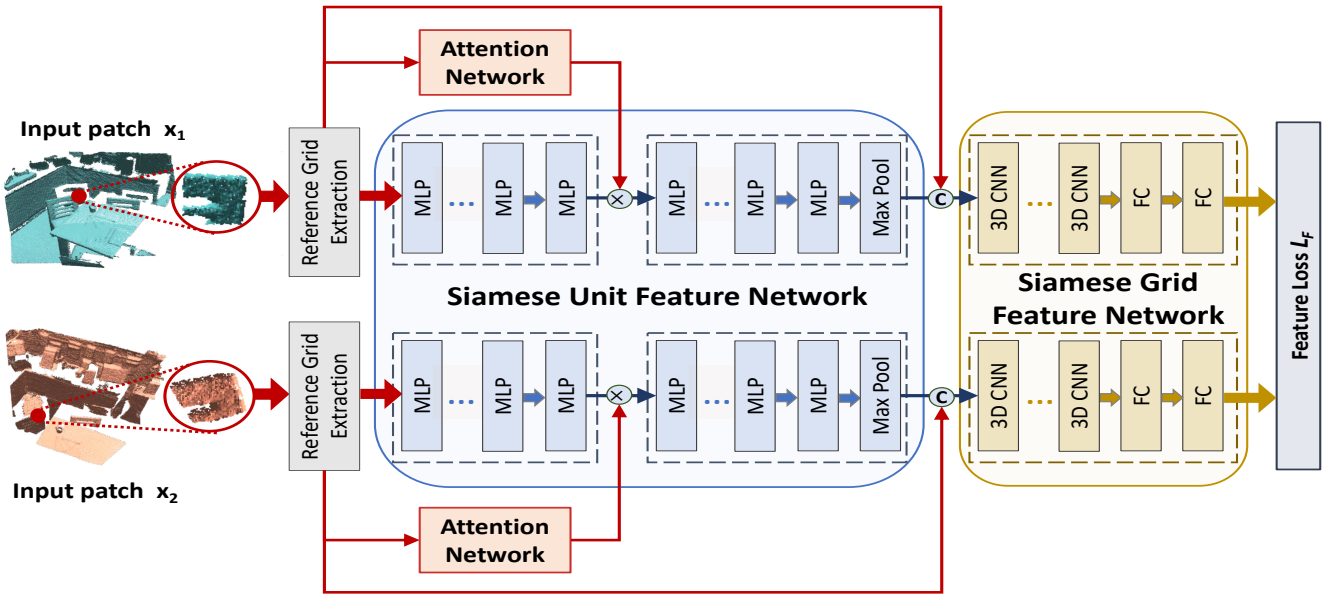

Figure 2: The framework of our proposed method. It consists of four parts: a reference grid extraction part, a siamese unit feature network, an attention network and a siamese grid feature network. Given a pair of keypoint patches from point could fragments, our proposed model first extracts a reference grid for each patch, and then learns a unit-wise feature from the reference grid using the MLP-based unit feature network. The MLP-based attention network is added upon the unit feature network to enhance the discriminability of our learned features. Taking the concatenation of the original reference grid and the unit-wise features as input, the siamese grid feature network learns a grid-wise feature as our final 3D point signature for each given patch.

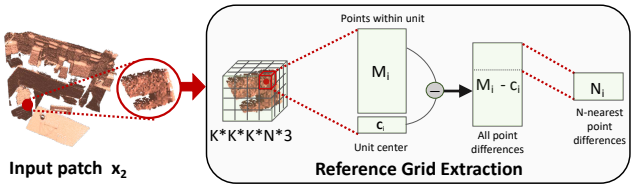

Figure 3: The pipeline of our reference grid extraction. Given a patch with points, we partition the patch into a $K \times K \times K$ grid. For each unit i in the grid, we extract all the M_i points fallen into the unit, and subtract the unit center c_i from all the M_i points per coordinate. Later, we take the n-nearest points N_i with (XYZ) coordinate differences as the values for the i-th unit. Finally, we have a $K \times K \times K \times N \times 3$ reference grid for each input patch.

Fig. 3. Given a patch with points (XYZ) coordinates, we first partition the patch into a $K \times K \times K$ grid. Then, for each unit i in the grid, we extract all the M_i points fallen into the unit, and subtract the unit center c_i from all the M_i points per (XYZ) coordinate. Later, we sort all the M_i points by the distances to the center c_i, from where we take the n-nearest points N_i with their (XYZ) coordinate differences as the values for the i-th unit. Finally, we have a $K^3 \times N \times 3$ reference grid $\boldsymbol{G_i}$ for each input patch $\boldsymbol{x_i}$.

Siamese Unit Feature Learning Network The purpose of the unit feature learning network is to discover the implicit relation within each unit of the reference grids for each input patch, so that our 3D point signature can be learned from the small unit region as a start. Inspired by the success of PointNet [22] on point cloud processing, we build the unit feature network with a pair of siamese networks, where each has seven MLP layers with channel sizes $\{8, 16, 32, 64, 64, 64, 128\}$, followed by a channel-wise max pooling layer. The output of the fourth MLP layers is multiplied by the attention features generated from the attention network before passing through the rest MLP layers. A LeakyReLU layer is added between each two MLP layers. The network parameters are shared between the siamese networks. After feeding a pair of reference grids $\boldsymbol{G_1}$ and $\boldsymbol{G_2}$ into the siamese unit feature network, we can get a pair of unit features $\boldsymbol{U_1}$ and $\boldsymbol{U_2}$ with a size of $K^3 \times 128$.

Attention Network In order to enhance the learning ability of the unit features, we introduce an attention network upon the unit feature learning network. Intuitively, the points closer to the unit center should have greater impact on the unit feature learning. Therefore, we compute the exponential of XYZ coordinate differences for points in each of the unit, then take the exponential reference grid as input for the attention network. Similar to the unit feature learning network, the attention network contains a pair

of siamese MLP-based networks with a channel size of $\{8, 16, 32, 64\}$. The attention network learns channel-wise attention features for each unit of the input reference grid. The outputs of the attention network are attention features with a size of $K^3 \times N \times 64$, implying the impact factors of each point within the unit on different channels. The attention features are multiplied by the outputs of the fourth MLP layer in the unit feature network as an enhancement.

Siamese Grid Feature Learning Network Before learning the grid-wise features, we concatenate the unit-wise feature $\boldsymbol{U_1}$ and $\boldsymbol{U_2}$ (generated from the unit feature network) with the original reference grid input pairs $\boldsymbol{G_1}$ and $\boldsymbol{G_2}$ as the input (size $K^3 \times 158$) for our grid feature learning network. In order to learn a 3D point signature from a larger perspective grid region, we develop our siamese grid feature learning network with three 3D convolutional layers. The channel sizes are $\{128, 256, 512\}$ with the kernel size of $3 \times 3 \times 3$ and the stride is 1. After a global average pooling layer, the pooled features are fed into three fully-connected layers with a neuron size of $\{512, 256, 256\}$. Every two network layers are connected by a LeakyReLU layer. Network parameters are shared between the siamese networks. Finally, the output of the last fully-connected layer is the 256-dimension 3D point signature for given keypoint (patch) input.

Network Training and Testing Let $|,|$ denotes the concatenation of two vectors, and $||\cdot||$ be the Euclidean distance between two feature vectors. P is the total number of training keypoint pairs. Given pairs of reference grids, we train all the networks, including the unit feature network *U*, attention network *A* and the grid feature network *F* simultaneously with contrastive loss

$$L_F = \frac{1}{2P} \sum (1-y) * ||F(|\boldsymbol{G_1}, \boldsymbol{U_1}|) - F(|\boldsymbol{G_2}, \boldsymbol{U_2}|)||^2 + y * \max(margin - ||F(|\boldsymbol{G_1}, \boldsymbol{U_1}|) - F(|\boldsymbol{G_2}, \boldsymbol{U_2}|)||, 0)^2. \quad (1)$$

We use ADAM optimizer to obtain the optimal network parameters with beta value $\beta = 0.5$. The learning rate is initialized as 0.001 and exponentially decayed after $200K$ training steps. The $margin$ is set to 1.0. When testing, given any keypoint in a scene point cloud, we extract its reference grid and pass in into our networks to obtain a 256-dimension point signature. We can match any given two keypoints by directly computing the Euclidean distances between their generated point signatures.

4. Experiments

In order to comprehensively evaluate the performance of our proposed method, we conduct two different experiments on large-scale RGB-D datasets, i.e., keypoint matching and geometry registration. In this section, we report the experiment settings and quantitative comparisons to state-of-the-art methods for both keypoint matching and geometry registration. We visualize some fragment alignments as the qualitative results to demonstrate the registration performance using our proposed signature with RANSAC algorithm.

4.1. Keypoint Matching

In order to provide a fair performance comparison, we conduct our experiments on the same benchmark organized by 3DMatch [33], which is constructed from SUN3D dataset, 7-Scenes dataset, RGB-D scenes V2 dataset and BundleFusion dataset with more than $100K$ RGB-D frames. The whole dataset has been split into non-overlap 46-scene training set and 8-scene testing set. We randomly sample $30K$ keypoint pairs from the depth images on training dataset with a ratio of $1:1$ for match and non-match pairs, and extract the points within $15cm$ radius around the sampled keypoints as patches. For testing, we use the same keypoint testing set provided by 3DMatch [33] that contains $10K$ pairs of keypoints sampled from the depth images of the 8-scene testing set, with $5K$ match pairs and $5K$ non-match pairs. Similarly, we extract the patch for each keypoint in the testing set from the point clouds fused by the depth images.

Given that the point cloud data is converted from depth images, the point cloud densities are various and as a consequence, the keypoint patch sizes vary from hundreds to thousands. Nevertheless, our introduced reference grid can perfectly eliminate this dynamic-density problem by providing a fixed-size $10 \times 10 \times 10$ grid for each patch. More importantly, our reference grid does not discard the density information, and actually includes it by computing the coordinate differences for points within each unit. We extract the 10-nearest neighbor points' coordinate differences for each unit. For those units with less than 10 points fallen into, we randomly replicate the points within the unit until the number of points reaches 10. The same reference grid extraction strategy is applied to all the keypoints for both training and testing.

We implemented our proposed framework using the popular deep learning platform Tensorflow [1], and trained it on the $30K$ pairs of training keypoints with a batch size of 30 for 20 epochs. It took around 12 hours on a desktop with Intel Xeon E5-2603 CPU and 501 NVIDIA Tesla K80 GPU. After training, we extract a 256-dimension point signature as the representation for each testing keypoint and match the point pairs based on the Euclidean distance calculated between their extracted signatures.

We report the false-positive rate (matching error rate) to measure the performance on keypoint matching, and our model obtains 30.0% matching error when recall reaches 95%. Moreover, we collect the publicly available results of

Table 1: Keypoint matching errors compared to state-of-the-art methods on the 3DMatch[33] testing dataset constructed from the SUN3D and 7-scenes datasets.

Type	Method	lower is better
		Error Rate(%)
Hand-crafted	Spin-Images [17]	83.7
	FPFH [24]	61.3
Voxel-based Learning	3DMatch (30×30×30) [33]	35.3
	3DMatch (10×10×10) [33]	51.0
Point-based Learning	PointNet [22] w contrastive loss	45.2
	PointNet++ [23] w contrastive loss	40.5
Ours	w/o Attention Network	**32.3**
	w Attention Network	**30.0**

state-of-the-art approaches (e.g. Spin-Images [17], FPFH [24], 3DMatch (30×30×30) [33]) from the 3DMatch website [1] for comparisons. Besides, we train a 3DMatch model on a dataset that contains voxelized patches with same size (10×10×10) as our proposed reference grid and calculate the matching errors. As we can see from Table 1, our proposed method outperforms the hand-crafted signatures and the voxel-based learning 3DMatch. Furthermore, our proposed method gets 20% higher accuracy than the 3DMatch model trained on the patches with same size (10×10×10).

In addition to the hand-crafted and voxel-based learning signatures, we also provide the keypoint matching performance using the popular point-based learning models PointNet [22] and PointNet++ [23]. We change the network structure of the PointNet a little bit to fit our matching problem. We only keep the classification network and discard the segmentation part. We randomly sample 1024 points within each patch and feed them into the modified PointNet. We train a siamese modified PointNet with a contrastive loss defined on the final fully-connected layer of the network with 256 neurons. Same modification is applied to the PointNet++ model. After training, we extract the 256-dimension features from the modified models as the learning point signatures for given testing keypoint. The keypoint matching errors are listed in Table 1. PointNet++ performs better than PointNet (40.5% V.S. 45.2% in error), since PointNet++ model takes the influence of the neighbor points into consideration. However, we can observe that the matching error of PointNet++ is still much higher than 3DMatch and our proposed method. The cause is that PointNet++ only learns the signature point-wisely, but fails to explore the grid-wise (or voxel-wise) relation over the patch.

To validate the effectiveness of the introduced attention network, we train our model without the attention network using the same experimental setting, and extract the point signature for each testing keypoint for matching. The matching error of our model without attention network is 32.3%, which is 2.3% higher than ours with the attention network. The error gap demonstrates that the attention network can indeed improve the discriminability of our learned 3D point signature for keypoint matching.

[1] http://3dmatch.cs.princeton.edu

4.2. Geometry Registration

In addition to the keypoint matching task, we verify our proposed method on geometry registration task, which is a challenging task to find the correspondences between any two fragments from the same scene. Following the same setting with 3DMatch [33], we randomly sample $5K$ keypoints from each testing point cloud fragments and extract the corresponding reference grids. After that, we utilize the same model trained on the keypoint training set (details in Section 4.1) to extract a 256-dimension point signature for each sampled keypoint.

We conduct experiments on the fused fragments constructed from some real-world scenes, including depth images in SUN3D dataset, 7-Scenes dataset, RGB-D scenes V2 dataset and BundleFusion dataset. We also report the registration results on the synthetic scene point cloud fragments, constructed from the depth images in the augmented ICL-NUIM dataset [13].

Assessment criteria Following the instruction provided by the latest work PPF-FoldNet [7], we validate our learned point signature by directly looking at the matching recall of the keypoints sampled from testing scenes. We compute the keypoint matching recall between match fragment pairs (**P**, **Q**) that have more than 30% overlap after being transformed by using their ground-truth transformation T^*. Let (p_i, q_i) represents a match keypoint pair from match fragment pair (**P**, **Q**) that are close to each other by applying Euclidean-distance-based nearest neighbor(NN) search on their corresponding point signatures $(f(p_i), f(q_i))$. Our matching recall for each scene is computed as

$$R = \frac{1}{M}\sum_{m=1}^{M}\mathbb{1}([\frac{1}{K}\sum_{i=1}^{K}\mathbb{1}(||p_i - T^*q_i||_2 < \tau_1)] > \tau_2), \tag{2}$$

where M is the total number of matched fragment pairs across the same testing scene, and K is the number of match keypoints after NN search on point signature space. We set $\tau_1 = 10cm$ and $\tau_2 = 0.05$ for all the testing scenes.

Real-world scene fragments The real-world scene dataset includes a total of 387 point cloud fragments categorized into 8 difference scenes, i.e. *homes*, *hotels*, *study-room* and *lab*. We compute the recalls using Eq. 2 and report the matching recalls on each testing real-world scene separately in Table 2, followed by an average recall of all testing scenes. As we can see from the table, the hand-crafted signatures Spin-Image [17] and FPFH [24] perform the worst among all the compared methods. It is reasonable since Spin-Image and FPFH were initially designed on the 3D mesh data with triangle connections among points. However, we only have points (coordinates) in each testing

Table 2: The recall comparisons of geometry registration before RANSAC on real-world scan SUN3D and 7-scenes datasets.

Type	Method	Matching Recall (%)								
		Kitchen	Home1	Home2	Hotel1	Hotel2	Hotel3	Studyroom	Lab	**Average**
Hand-crafted	Spin-Images [17]	19.4	39.7	36.5	18.1	20.2	31.5	5.5	10.4	22.7
	FPFH [24]	30.6	58.3	46.6	26.1	32.7	50.0	15.4	27.3	35.9
Voxel-based Learning	3DMatch [33]	57.5	73.7	70.7	57.1	44.2	63.0	56.2	54.6	59.6
Point-based Learning (w hand-crafted features)	PPFNet [8]	**89.7**	55.8	59.1	58.0	57.7	61.1	53.4	**63.7**	62.3
	PPF-FoldNet [7] (unsupervised)	78.7	76.3	61.5	68.1	**71.1**	**94.4**	62.0	62.3	71.8
Ours	w/o Attention Network	68.8	77.6	78.3	73.4	63.4	66.7	59.2	58.9	68.3
	w Attention Network	73.5	**85.3**	**81.7**	**77.4**	69.2	75.9	**65.1**	62.3	**73.8**

Table 3: The recall comparisons of geometry registration before RANSAC on the synthetic ICL-NUIM dataset.

Method	Matching Recall (%)				
	Liv. 1	Liv. 2	Off. 1	Off. 2	**Avg.**
Spin-Images [17]	15.9	38.2	22.1	49.7	31.5
FPFH [24]	21.4	35.7	32.9	42.1	33.0
3DMatch [33]	26.0	39.7	44.2	53.3	40.8
w/o Attention Network	26.7	45.2	49.6	59.4	45.2
w Attention Network	32.1	53.3	56.3	67.8	52.4

scene fragment. The voxel-based learning 3DMatch obtains higher recalls over all the categories when compared to the hand-crafted ones. We also present the matching recalls of two recent point-based learning methods, i.e. PPFNet [8] and PPF-FoldNet [7]. They achieve much better matching performance than the hand-crafted and voxel-based learning approaches due to the leverage of some auxiliary hand-crafted feature when training their models.

For our proposed method, we provide the matching recalls of the models trained with the attention network and without the attention network. The model trained with the attention network attains the best average keypoint matching recall 73.8% across the testing dataset. We also observe that our proposed method performs best on category *Home1*, *Home2*, *Hotel1* with large recall margins. The noticeable 5.5% improvement on average recall implies the effectiveness of our introduced attention network on the point signature learning.

Synthetic scene fragments Besides the evaluation on point cloud fragments in real-world scenes, we also conduct the geometry registration task on the synthetic scan scenes in augmented ICL-NUIM dataset [13]. For fair comparison, we test our model on the well-constructed fragments provided by the authors of 3DMatch [33], which contains 207 fragments categorized into four different scenes (e.g. *livingroom1*, *livingroom2*, *office1* and *office2*), and each of the fragment has $5K$ sampled keypoints. In order to find the domain adaption ability of our proposed model, we directly extract the 256-dimension point signature for each synthetic scene sampled keypoint using the model trained on the real-world scene training dataset (same model as the one mentioned in Section 4.1) .

Matching recalls are computed directly on the extracted point signatures for each of the four synthetic scenes using Eq. 2. As reported in Table 3, our proposed point signature can obtain 32.1%, 53.3%, 56.3%, 67.8% recalls on *livingroom1*, *livingroom2*, *office1* and *office2*, respectively. The average recall is 52.4%. For comparison, we also calculate the matching recalls using hand-crafted signatures Spin-Images [17] and FPFH [24], voxel-based learning signatures 3DMatch [33]. Similarly, we use the 3DMatch model trained on the real-world scan scene training dataset to extract point signatures for domain adaption comparison. The performance improvement demonstrates a greater domain adaption potential of our proposed model.

4.3. Fragment Alignment

In Section 4.2, we evaluate our proposed mode in the geometry registration task with quantitative keypoint matching recalls. In this section, we further validate the proposed point signature with some qualitative fragment alignment examples. Generally, the more robust a point signature is, the more correct alignment it would produce with RANSAC algorithm.

We leverage the $5K$ sampled keypoints (in Section 4.2) from each point cloud fragment, and extract their point signatures from our trained model. After that, we can get match point pairs by applying NN search on the point signature space, and then get the estimated transformation between any two point cloud fragments using classic RANSAC algorithm. Finally, the estimated transformation can be used to align the two fragments. We visualize some examples of the aligned fragment pairs as the qualitative evaluation of our learned point signature. Moreover, we pick some alignment results based on 3DMatch signature for comparison. All the fragment pairs are aligned by multiplying the points (coordinates) with the estimated transformation matrix generated by applying RANSAC on match keypoint pairs collected with signatures.

In Fig. 4, we provide three pairs of point cloud fragments from *living room* in synthetic ICL-NUIM dataset, *hotel* and *studyroom* in real-world SUN 3D dataset. All of them are the challenging cases we have found from the testing datasets. 3DMatch fails to get a correct transformation on the case that only a vey small common part exist between the two fragments, like the *livingroom* example. For the *hotel* fragment pair, our proposed signature is able to get a more correct transformation that perfectly aligns the

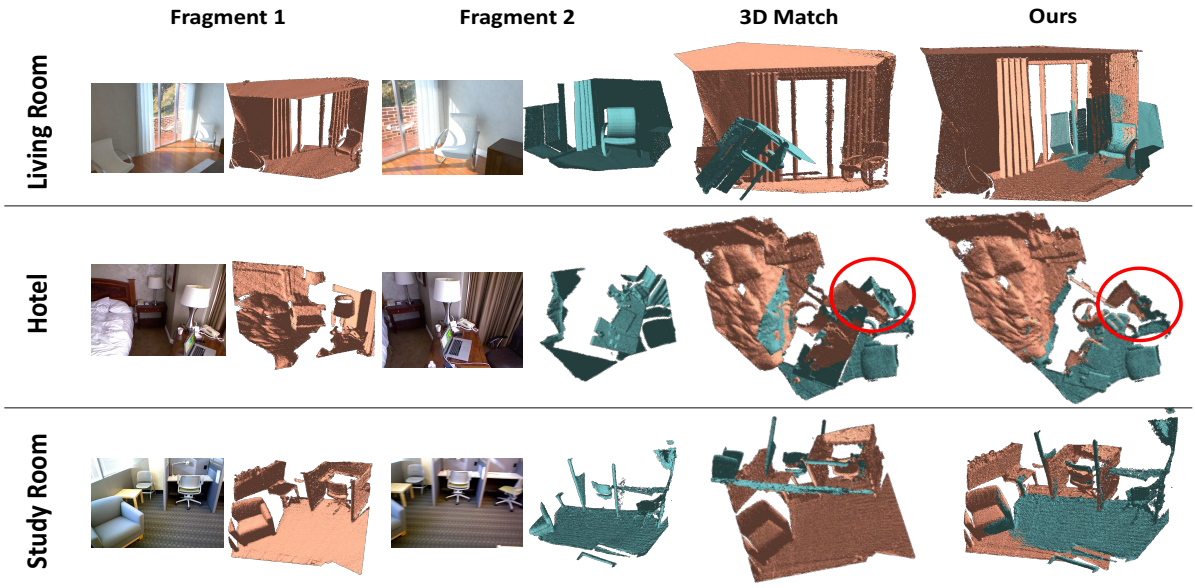

Figure 4: Examples of some challenging fragment alignment cases in the real-world scenes and synthetic scenes. All the alignment results are obtained after applying the RANSAC algorithm. We can observe that our point signature can successfully handle those challenges but 3DMatch fails.

Table 4: Keypoint matching errors using our proposed models trained with different components.

Components	lower is better
	Error Rate(%)
grid feature network	47.0
unit feature network	40.7
unit feature network + attention network	38.0
unit feature network + grid feature network	32.3
All combined	**30.0**

two fragments. In the *studyroom* example, 3DMatch cannot distinguish the two chairs in the fragment2, confusing the RANSAC to compute an incorrect transformation. Our proposed method successfully handles those challenges.

4.4. Ablation Study

Furthermore, we conduct the ablation study on our proposed framework by training the model with different components, e.g., the unit feature network, the attention network and the grid feature network. When removing the components, there could be some incompatibilities between contrastive training loss and the feature outputs. We adjust the network structure slightly so that all the compared models in the ablation study can be trained with the same contrastive loss, and they all generate a 256-dimension point signature for any given keypoint.

Following the same experiment setting in Section 4.1, we compute the matching errors on the 3DMatch keypoint matching testing dataset with the point signatures generated by different models. All the keypoint matching errors are listed in Table 4. If we train a grid feature network and unit feature network independently, we can get 47% and 40.7% matching errors, respectively. It implies that the unit feature network actually contributes more than the grid feature network for the point signature learning. With the involvement of the unit feature network, the grid feature network can learn a point signature to get a much lower error (32.3%). Additionally, the introduced attention network can improve $\approx 2\%$ matching accuracy compared to the models trained without the attention network.

5. Conclusion

In this paper, we tackle the challenging 3D point signature problem by learning a robust point signature from raw point clouds without any precomputed pairwise features or normals among points. In order to better describe the patch density and neighbor points relation to the keypoint, we design a reference grid for each keypoint patch that contains the neighbor point XYZ differences from the grid centers. Specifically, we develop a novel siamese MLP-base unit feature network to learn the neighbor relation within each unit of the reference grid, followed by a 3D CNN-based grid feature network to capture the grid-wise characteristics. Moreover, an attention network is introduced on the unit feature network to enhance the discriminability of the learned 3D point signature. The experimental results demonstrate that our proposed signature outperforms the state-of-the-art signatures on keypoint matching and geometry registration. More importantly, our learned 3D point signature can handle the alignment challenges with small overlap between fragments.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[3] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. In *Computer Graphics Forum*, volume 34, pages 13–23. Wiley Online Library, 2015.

[4] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016.

[5] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032, 2015.

[6] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[7] H. Deng, T. Birdal, and S. Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[8] H. Deng, T. Birdal, and S. Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong. 3d deep shape descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2015.

[10] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics (TOG)*, 25(1):130–150, 2006.

[11] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[12] K. Guo, D. Zou, and X. Chen. 3d mesh labeling via deep convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 35(1):3, 2015.

[13] A. Handa, T. Whelan, J. McDonald, and A. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.

[14] J. Hu, J. Lu, and Y.-P. Tan. Deep transfer metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 325–333, 2015.

[15] H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. G. Kim, and E. Yumer. Learning local shape descriptors with view-based convolutional networks. *arXiv preprint arXiv:1706.04496*, 2017.

[16] Q. Huang, W. Wang, and U. Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[17] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[21] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[22] 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*, pages 652–660, 2017.

[23] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[24] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[25] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3384–3391. IEEE, 2008.

[26] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Learning informative point classes for the acquisition of object model maps. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 643–650. IEEE, 2008.

[27] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[28] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang. Contextual part analogies in 3d objects. *International Journal of Computer Vision*, 89(2-3):309–326, 2010.

[29] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou. Tangent convolutions for dense prediction in 3d. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[30] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] Z. J. Yew and G. H. Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *European Conference on Computer Vision*, pages 630–646. Springer, 2018.

[32] L. Yi, H. Su, X. Guo, and L. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *arXiv preprint arXiv:1612.00606*, 2016.

[33] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.

[34] E. Zhang, K. Mischaikow, and G. Turk. Feature-based surface parameterization and texture mapping. *ACM Transactions on Graphics (TOG)*, 24(1):1–27, 2005.

[35] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[36] J. Zhu, F. Zhu, E. K. Wong, and Y. Fang. Learning pairwise neural network encoder for depth image-based 3d model retrieval. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1227–1230. ACM, 2015.