Improving RGB-D Point Cloud Registration by Learning Multi-scale Local Linear Transformation

Ziming Wang[1*], Xiaoliang Huo[1*], Zhenghao Chen[2], Jing Zhang[1], Lu Sheng[1†], and Dong Xu[3]

[1] School of Software, Beihang University
[2] School of Electrical and Information Engineering, The University of Sydney
[3] Department of Computer Science, The University of Hong Kong
{by1906050,huoxiaoliangchn,lsheng}@buaa.edu.cn

Abstract. Point cloud registration aims at estimating the geometric transformation between two point cloud scans, in which point-wise correspondence estimation is the key to its success. In addition to previous methods that seek correspondences by hand-crafted or learnt geometric features, recent point cloud registration methods have tried to apply RGB-D data to achieve more accurate correspondence. However, it is not trivial to effectively fuse the geometric and visual information from these two distinctive modalities, especially for the registration problem. In this work, we propose a new Geometry-Aware Visual Feature Extractor (GAVE) that employs multi-scale local linear transformation to progressively fuse these two modalities, where the geometric features from the depth data act as the geometry-dependent convolution kernels to transform the visual features from the RGB data. The resultant visual-geometric features are in canonical feature spaces with alleviated visual dissimilarity caused by geometric changes, by which more reliable correspondence can be achieved. The proposed GAVE module can be readily plugged into recent RGB-D point cloud registration framework. Extensive experiments on 3D Match and ScanNet demonstrate that our method outperforms the state-of-the-art point cloud registration methods even without correspondence or pose supervision.

Keywords: Point cloud registration, geometric-visual feature extractor, local linear transformation

1 Introduction

Point cloud registration [15,5,8,19,3,21,1,38] is a task to estimate geometric transformation, such as rotation and translation, between two point clouds. By applying the geometric transformation, we can merge the partial scans from two

* indicates equal contributions
† Lu Sheng is the corresponding author, e-mail: lsheng@buaa.edu.cn

views of the same 3D scene or object into a complete 3D point cloud, which is a key component of numerous tasks in the community of robotics and AR/VR and also plays an essential role on understanding the whole environment.

The common approach to point cloud registration relies on two processes: (1) correspondence extraction and (2) geometric model fitting, where accurate correspondence is the key for reliable model fitting. The recent 3D deep learning techniques [9,10,8,15,16,19,39,12] outperform the traditional methods [5,31] by finding more accurate correspondence based on learnable geometric features [10,16], or further combining the model fitting process into an end-to-end learning framework [19,15,16,8]. However, the geometric features from 3D points are still less discriminative in comparison to visual features from the RGB images. Thanks to the rapid popularization of RGB-D cameras, it becomes promising to collect the RGB-D data for extracting more reliable correspondence, such that both geometric and visual consistencies can be well examined between two views. A couple of learning based works [15,16] belong to this line of work, which achieve superior registration performance even without ground-truth poses or correspondence as their supervision information. However, UR&R [15] just uses RGB images for correspondence estimation, while BYOC [16] relies on pseudo-correspondence from RGB images to train the geometric correspondence. Thus both methods [15,16] do not fully leverage the complementary visual and geometric information. Moreover, according to our experiments (see Section 4), we can only achieve marginal gains by simply concatenating RGB-D data as the input for correspondence estimation in UR&R [15]. A possible explanation that it is hard to fully exploit the geometry clues by using the CNN networks due to the intrinsic difference between the geometric and visual features.

To this end, we propose a Geometry-Aware Visual Feature Extractor (GAVE) that can generate distinctive but comprehensive geometric-visual features from RGB-D images, which facilitates reliable correspondence estimation for better point cloud registration. This module can be readily used to replace the feature extractor in UR&R [15], and significantly improve the point cloud registration performance even trained in an unsupervised manner[1]. To be specific, in the GAVE module, we propose a Local Linear Transformation (LLT) module, where the geometric features (extracted from the geometric feature extractor) act as the guided signal and are converted as point-wise linear coefficients to enhance the visual features (extracted from the visual feature extractor), through point-wise linear transformation. Moreover, to enhance the content awareness of the transformation with respect to the input depth image, we borrow the idea from the edge-aware image enhancement method [17], which employs the Bilateral Grid and an edge-aware guidance map (both are estimated from the depth image) to generate our content-aware linear coefficients. Note that this LLT module is applied in the GAVE module in a multi-scale fashion, which thus enriches the scale awareness of the generated visual-geometric features.

[1] As shown in the ablation study, GAVE module can also be applied into the supervised pipelines.

More specifically, the proposed LLT module can be viewed as multi-scale dynamic convolutions over visual features that are guided by the geometric clues, which offers more descriptive and complementary combination between visual and geometric features than the common used operations such as concatenation, summation or product. Since the goemetric feature can represent local geometric structure and indicate local orientations, it is easier for dynamic convolution-based fusion network to learn how to better project visual features into the new feature space where the projected features are robust to geometric changes, which is crucial for registration. To our best knowledge, it is the first work that applies a dynamic convolution-based fusion strategy in RGB-D point cloud registration, whose design is tailored to the nature of this particular task.

Our Geometry-Aware Visual Feature Extractor is trained in an end-to-end manner together with the subsequent correspondence estimation and differentiable geometric model fitting modules, *e.g.*, those from UR&R [15]. The state-of-the-art results are achieved on the standard point cloud registration benchmark dataset ScanNet [11] with the models respectively trained based on the ScanNet [11] and 3D Match [39] datasets, which clearly outperform the existing point cloud-based supervised baselines and RGB-D-based unsupervised methods.

2 Related Work

2.1 3D Feature Extractors.

To extract the useful 3D features for various 3D vision tasks, early methods adopted the hand-crafted statistic-based strategies [5,27,4,32] to discover local 3D geometries. With the recent success of deep learning techniques, many learning-based 3D feature extraction methods [12,15,16,9,10,6] have been proposed. While some of them are proposed for extracting the features from point clouds [9,10,30,2], our methods are inspired more from those methods that extract the features from RGB-D images/videos [11,39,29,34,33]. However, most existing geometric-visual feature extractors just simply combine the features respectively from RGB images and depth maps without carefully considering how to exploit their correlation.

2.2 Bilateral Feature Fusion

Several methods [22,17,18,36,37,7] have conducted feature fusion in a bilateral manner. Particularly, the works [17,36,37] produce the edge-aware affine color transformation by using the Bilateral Grid. Inspired by those methods, we also develop the content-aware local linear coefficients through the Bilateral Grid, which act as the geometry-guided convolution kernels to transform the visual features.

2.3 3D Point Cloud Registration

The earlier 3D point cloud registration methods extracted point cloud features and then align them with robust model fitting technologies [24,13,27,4,32,14,10].

Some learning-based methods [10,19,8] leverage the extra ground-truth poses to learn better geometric features from point clouds. However, it is not trivial to collect such ground-truth annotations. Recently, the unsupervised learning methods, such as UR&R [15] and BYOC [16], enforce cross-view geometric and visual consistency to implicitly supervise the training of registration. But the features used for correspondence extraction are either directly based on the RGB data [15] or trained by the pseudo-correspondence labels from the visual correspondences [15], where the extraction of visual and geometric clues are usually independent without effectively exploiting their correlation. Our work is inspired by [15,16], but would like to explore more reliable geometry-aware visual features for more robust registration. In contrast to those existing methods adopting the fusion operations such as concatenation, summation and attention [35], our newly proposed LLT module explores the correlation between visual and geometric information by using multi-scale dynamic convolutions over visual features, whose kernels are guided by the geometric clues.

3 Methodology

In this work, we propose a Geometry-Aware Visual Feature Extractor (GAVE) to learn distinctive and comprehensive geometric-visual features. Specifically, given each RGB-D image, the GAVE module extracts the visual features and geometric features in a parallel way, and then we densely apply the newly proposed Local Linear Transformation (LLT) module in a multi-scale fashion to progressively fuse the features from these two modalities. Therefore, a pair of RGB-D images $\{\mathbf{I}_R, \mathbf{I}_T\}$ ($\mathbf{I}_R$ as the reference RGB-D image, $\mathbf{I}_T$ as the target RGB-D image) can be encoded as a pair of geometric-visual features $\{\mathbf{F}_R, \mathbf{F}_T\}$, which are then inputted into a correspondence estimation module to calculate the correspondence. The set of captured correspondence is finally used in a geometric model fitting module (*e.g.*, a differentiable alignment module in UR&R [15]), together with the point clouds $\mathcal{P}_R$ and $\mathcal{P}_T$ converted from $\mathbf{I}_R$ and $\mathbf{I}_T$, to produce the rotation matrix $\mathbf{R}_{R\to T} \in \mathbb{R}^{3\times3}$ and the translation vector $\mathbf{t}_{R\to T} \in \mathbb{R}^{3\times1}$ from the reference RGB-D image to the target RGB-D image. In our work, we adopt the correspondence estimation and differentiable alignment modules from UR&R [15] in addition to our GAVE module, thus the whole registration framework can be trained in an end-to-end unsupervised learning manner. The overall framework is shown in Figure 1.

3.1 Overview of Our Geometry-Aware Visual Feature Extractor

In our Geometry-Aware Visual feature Extractor, we have two parallel subnetworks, namely a visual feature extractor and a geometric feature extractor respectively, to extract the visual and geometric features from a RGB-D image. The visual feature extractor contains 2 dilated convolution blocks to enlarge the receptive fields that describes the visual contents. The geometric features after the last 2 convolution blocks are converted into a set of Bilateral Grids [17],

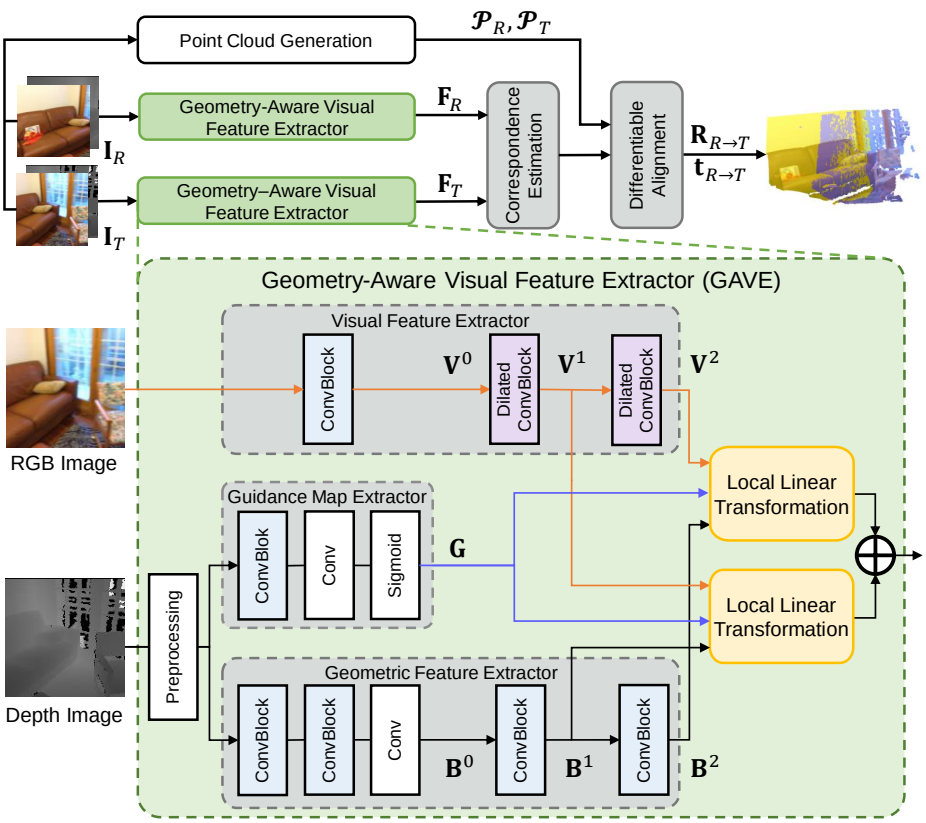

Fig. 1. The overview of our Geometry-Aware Visual Feature Extractor (GAVE) based framework. We first generate the multi-scale visual features, multi-scale geometric features and the guidance map, respectively. Then, we fuse the extracted visual and geometric features by using our proposed Local Linear Transformation (LLT) modules with the learned guidance map to produce the intermediate visual-geometric features. The intermediate features from two different scales are then averaged to generate the final visual-geometric features. Once we obtain the pair of the visual-geometric features ($\mathbf{F}_R$, $\mathbf{F}_T$) from the reference RGB-D image and the target RGB-D image, we can then perform the matching and registration operations to produce the rotation matrix and the translation vector by using the correspondence generation and the differentiable alignment module in [15]. The details of our proposed LLT module and the basic *ConvBlock* and *Dilated ConvBlock* modules will be illustrated in Figure 2.

which act as the source of the local linear coefficients for the proposed Local Linear Transformation modules. In addition, based on depth image, we also produce an edge-aware guidance map that further helps to interpolate geometry-dependent linear coefficients from the predicted Bilateral Grids. Since then, the LLT module progressively applies the extracted guidance map to slice the set of Bilateral Grids, by which the resultant local linear coefficients are employed

to transform the visual features, which are extracted after the last two dilated convolutional blocks in the visual feature extractor. Since our GAVE module adopts a multi-scale fusion strategy, we can produce the final visual-geometric features by averaging the outputs from both LLT modules. More details of each component will be respectively introduced in the following sections.

3.2 Visual Feature Extractor

We apply dilated convolutions to enlarge the receptive fields in the visual feature extractor. Specifically, the visual feature extractor at first extracts an initial visual feature map $\mathbf{V}^0 \in \mathbb{R}^{H\times W\times D_c}$ by using a $ConvBlock(64, 3, 1)$ operation (*i.e.,* $D_c = 64$). H and W are the height and width of the input RGB image. Then, two dilated convolution blocks $Dilated\,ConvBlock(64, 3, 1, 2)$ are stacked thereafter, where visual feature maps $\mathbf{V}^1 \in \mathbb{R}^{H\times W\times D_c}$ and $\mathbf{V}^2 \in \mathbb{R}^{H\times W\times D_c}$ are generated from each block, as the sources for multi-scale feature fusion. There are no downsampling operations in this module, thus the output visual feature maps have the same spatial size as the input image. Note that the definitions of $ConvBlock(N, K, S)$ and $Dilated\,ConvBlock(N, K, S, d)$ are depicted in Figure 2(b), where N is the number of output channel, K is the kernel size, S is the stride and d refers to the dilation factor.

3.3 Geometric Feature Extractor

The input depth image is at first normalized to $[0, 1)$ through some linear normalization operations with a `sigmoid` function. Since raw depth image may contain holes due to sensor's systematic errors, thus in the pre-processing step, we also apply the Joint Bilateral Filtering (JBF) method [26] to fill the depth holes with the aid of the corresponding RGB image.

Once we produce the normalized depth map, we first encode it by using a stack of convolution operations (*i.e.,* $ConvBlock(32, 3, 2)$, $ConvBlock(256, 3, 2)$ and $Conv(768, 3, 2)$[2]) to generate an initial down-scaled geometric feature map $\mathbf{B}^0 \in \mathbb{R}^{(H/8)\times(W/8)\times D_d}$ (we use $D_d = 768$ in this work, so as to match the size of the visual features, which will be explained in Section 3.5). Since then, we use two $ConvBlock(768, 3, 1)$ modules to respectively generate two geometric feature maps representing two different scales. Each Bilateral Grid is reshaped from each geometric feature map as $\mathbf{B}^i \in \mathbb{R}^{(H/8)\times(W/8)\times(D_d/n_{grid})\times n_{grid}}$, where $i = 1, 2$, and n_{grid} is the depth of the Bilateral Grid (in this work we set $n_{grid} = 3$ for balancing the efficiency and effectiveness). Please refer to [17] for more details about Bilateral Grid.

3.4 Guidance Map Extractor

In order to provide the content-structural information when generating the local linear coefficients, we define a guidance map by using a point-wise nonlinear

[2] $Conv(N, K, S)$ is a standard convolution operation with the output channel N, the kernel size K and the stride size S.

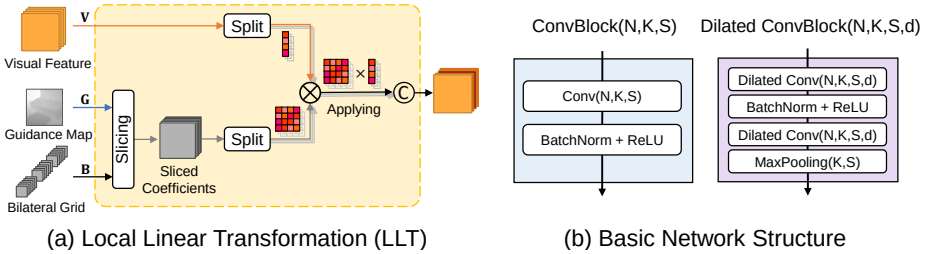

Fig. 2. For our "Local Linear Transformation" in (a), we first produce the sliced coefficients from the guidance map and Bilateral Grid by using the slicing operation in [17]. Then, at each position, the sliced coefficient matrix will be split in to group-wise coefficient matrix. To produce the transformed feature at each group, we then perform the applying operation (*i.e.,* the linear transformation) between each group-wise coefficient and each group-wise visual feature, which is also split from the previously learned visual feature. Finally, we generate the final output feature by using the channel-wise concatenation operation on these transformed features from all groups. The basic module is shown in (b) "Basic Network Structure". "$Conv(N, K, S)$" represents the convolution operation with the output channel, the kernel size and the stride as N, $K \times K$ and S, respectively. "d" in "$Dilated\,Conv(N, K, S, d)$" refers to the dilation parameter of the dilated convolution operation.

transformation on the depth map. Specifically, we input the normalized and hole-filled depth map and then directly employ several convolution operations (*i.e.,* $ConvBlock(3, 3, 1)$ and $Conv(1, 3, 1)$) and a `sigmoid` activation function to produce a learned guidance map $\mathbf{G} \in \mathbb{R}^{H \times W \times 1}$, which preserves the piece-wise smoothness as well as discontinuity presented in the depth image.

3.5 Multi-scale Local Linear Transformation

In order to effectively fuse the visual and geometric clues, we progressively apply the local linear transformation to learn the visual-geometric features in a multi-scale manner. To be specific, in each LLT module, we would like to *slice* the generated Bilateral Grids to produce the content-aware local linear coefficients, and then *apply* the sliced coefficients to transform the visual features. For the sake of efficiency, we also split the local linear transformation into several groups evenly along the channel dimension, and then concatenate these group-wise outputs as the final visual-geometric features.

Slicing The slicing operation is performed between a Bilateral Grid $\mathbf{B}$ (*i.e.,* $\mathbf{B}^1$ or $\mathbf{B}^2$) and the guidance map $\mathbf{G}$. At each spatial location in the guidance map, we use its spatial coordinates and the value at that location to sample nearest points in the Bilateral Grid, and then bilinearly interpolate the sampled coefficients to eventually generate the sliced linear coefficients. Therefore, the sliced coefficients become a tensor $\tilde{\mathbf{A}} \in \mathbb{R}^{H \times W \times (D_d / n_{grid})}$. Note that we split the local linear transformation into n_{group} ($n_{group} = 16$ in this work) groups, namely

the sliced coefficients $\tilde{\mathbf{A}}$ can be reshaped as $\mathbf{A} \in \mathbb{R}^{H\times W\times D_c\times (D_c/n_{group})}$. Note that when $D_c = 64$ and $D_d = 768$, $n_{grid} = 3$ and $n_{group} = 16$, $\tilde{\mathbf{A}}$ and $\mathbf{A}$ have the same number of elements. In this way, the context of the sliced coefficient tensor will be conditioned on the content structure from the guidance map, and such slicing operation is more computational-friendly than `softmax`-based interpolation.

Apply After evenly splitting the sliced linear coefficients $\mathbf{A}$ to produce $\mathbf{A}^{(g)} \in \mathbb{R}^{H\times W\times (D_c/n_{group})\times (D_c/n_{group})}, g = 1, \ldots, n_{group}$, the final local linear transformation can be obtained by first using a point-wise transformation and then using a channel-wise concatenation, such as

$$\mathbf{F} = \mathop{\Big\|}_{g=1}^{n_{group}} \mathbf{A}^{(g)} \otimes \mathbf{V}^{(g)} \tag{1}$$

where $\mathop{\Big\|}_{g=1}^{n_{group}}$ means channel-wise concatenation among n_{group} linearly transformed group-wise features, $\otimes$ indicates the matrix multiplication operation at every spatial position. $\mathbf{V}^{(g)} \in \mathbb{R}^{H\times W\times (D_c/n_{group})}$ is the g-th group of $\mathbf{V}$, which is evenly split from $\mathbf{V}$ along the channel dimension. Moreover, $\mathbf{F} \in \mathbb{R}^{H\times W\times D_c}$ has the same size as the visual feature $\mathbf{V}$. Note that we take the LLT module from one scale as an example for better illustration. We omit the superscript i (*i.e.,* the scale index) in $\mathbf{V}$, $\mathbf{V}^{(g)}$ and $\mathbf{F}$ for brevity, as the LLT module from each scale shares the same process. Thus, our local linear transformation-based fusion method inherently takes advantage of both modalities and provide a more flexible fusion strategy than simple concatenation or summation operations.

Multi-scale Fusion The fused visual-geometric features $\mathbf{F}^i \in \mathbb{R}^{H\times W\times D_c}, i = 1, 2$ in both scales are then averaged at the end of the GAVE module, so as to fulfill the multi-scale awareness of the features, which is essential for correspondence estimation in point cloud registration.

3.6 Correspondence, Registration and Objective Functions

Correspondence and Registration After feeding two RGB-D images $(\mathbf{I}_R, \mathbf{I}_T)$ to our GAVE extractor to produce the visual-geometric feature pairs ($\mathbf{F}_R$, $\mathbf{F}_T$), we can then perform the following correspondence estimation and differentiable alignment operations. Specifically, we first follow the work in [15] to compute the top-k (we set $k = 400$ in this work) correspondence pairs and then employ such correspondence pairs to estimate the rotation matrix $\mathbf{R}_{R\to T}$ and the translation vector $\mathbf{t}_{R\to T}$ by using the differentiable alignment module [15].

Objective Function As proposed by UR&R [15], we also apply the photometric, depth and correspondence consistencies to train the whole RGB-D point

Table 1. Pairwise registration errors on the ScanNet [11] dataset. We report the mean and median errors in terms of rotation error (°), translation error (mm), and Chamfer distance (cm). Features for correspondence estimation may come from visual or geometric/3D modality. The training set can be 3D Match [39] or ScanNet [11]. "Sup" means training with ground-truth pose supervision.

Methods	Train Set	Sup	Features		Rotation		Translation		Chamfer		FMR
			Visual	3D	Mean	Med.	Mean	Med.	Mean	Med.	
SIFT [27]	N/A		✓		18.6	4.3	26.5	11.2	42.6	1.7	-
SuperPoint [14]	N/A		✓		8.9	3.6	16.1	9.7	19.2	1.2	-
FCGF [10]	N/A			✓	9.5	3.3	23.6	8.3	24.4	0.9	-
BYOC [16]	3D Match		✓	✓	7.4	3.3	16.0	8.2	9.5	0.9	-
DGR [8]	3D Match	✓		✓	9.4	1.8	18.4	4.5	13.7	0.4	-
3D MV Reg [19]	3D Match	✓		✓	6.0	1.2	11.7	2.9	10.2	0.2	-
UR&R [15]	3D Match		✓		4.3	1.0	9.5	2.8	7.2	0.2	0.78
UR&R (RGB-D)	3D Match		✓	✓	3.8	1.1	8.5	3.0	6.5	0.2	0.78
Ours	3D Match		✓	✓	**3.0**	**0.9**	**6.4**	**2.4**	**5.3**	**0.1**	**0.87**
BYOC [16]	ScanNet		✓	✓	3.8	1.7	8.7	4.3	5.6	0.3	-
UR&R [15]	ScanNet		✓		3.4	0.8	7.3	2.3	5.9	0.1	0.85
UR&R (RGB-D)	ScanNet		✓	✓	2.6	0.8	5.9	2.3	5.0	0.1	0.91
Ours	ScanNet		✓	✓	**2.5**	**0.8**	**5.5**	**2.2**	**4.6**	**0.1**	**0.94**

cloud registration framework. The photometric consistency is measured by comparing the target image with the differentiably rendered reference image, according to the estimated rotation and translation parameters. The depth consistency is similar to the photometric consistency, but it compares the depth value instead. Correspondence consistency directly measures the matching errors between the corresponded points. Please refer to [15] for more details.

4 Experiments

4.1 Datasets and Experimental Setup

Datasets We follow UR&R [15] and adopt the large-scale indoor RGB-D dataset ScanNet [11] for evaluating our proposed GAVE module. Specifically, there are 1513 scenes in the ScanNet dataset [11] and each scene contains both RGB-D images and their ground-truth camera poses. We use its original training/testing split, which respectively contain 1045 and 312 scenes. In addition, as in [15], we also provide more evaluation results, in which we train our model based on another smaller point cloud dataset 3D Match [39] with 101 real-world indoor scenes and then evaluate the learnt model on the ScanNet dataset. Each scene in 3D Match also provides RGB-D images and point clouds data.

Evaluation Metrics We adopt the evaluation metrics, *i.e.,* the rotation error, the translation error and the Chamfer distance, as used in UR&R [15]. We report both the mean and the median values for these three error metrics in Section 4.2 and Section 4.3. In addition, we report the registration accuracy, *i.e.*, the rotation

Table 2. Pairwise registration accuracies on the ScanNet [11] dataset. We report the rotation accuracy with different angles (*i.e.,* 5°, 10° and 45°), the translation accuracy with different lengths (*i.e.,* 5cm, 10cm and 25cm) and the Chamfer accuracy with different metric distances (*i.e.,* 1mm, 5mm and 10mm).

Methods	Train Set	Sup	Features		Rotation			Translation			Chamfer		
			Visual	3D	5	10	45	5	10	25	1	5	10
SIFT [27]	N/A		✓		55.2	75.7	89.2	17.7	44.5	79.8	38.1	70.6	78.3
SuperPoint [14]	N/A		✓		65.5	86.9	96.6	21.2	51.7	88.0	45.7	81.1	88.2
FCGF [10]	N/A			✓	70.2	87.7	96.2	27.5	58.3	82.9	52.0	78.0	83.7
BYOC [16]	3D Match		✓	✓	66.5	85.2	97.8	30.7	57.6	88.9	54.1	82.8	89.5
DGR [8]	3D Match	✓		✓	81.1	89.3	94.8	54.5	76.2	88.7	70.5	85.5	89.0
3D MV Reg [19]	3D Match	✓		✓	87.7	93.2	97.0	69.0	83.1	91.8	78.9	89.2	91.8
UR&R [15]	3D Match		✓		87.6	93.1	98.3	69.2	84.0	93.8	79.7	91.3	94.0
UR&R (RGB-D) [15]	3D Match		✓	✓	87.6	93.7	98.8	67.5	83.8	94.6	78.6	91.7	94.6
Ours	3D Match		✓	✓	**93.4**	**96.5**	**98.8**	**76.9**	**90.2**	**96.7**	**86.4**	**95.1**	**96.8**
BYOC [16]	ScanNet		✓	✓	86.5	95.2	99.1	56.4	80.6	96.3	78.1	93.9	86.4
UR&R [15]	ScanNet		✓		92.7	95.8	98.5	77.2	89.6	96.1	86.0	94.6	96.1
UR&R (RGB-D) [15]	ScanNet		✓	✓	94.1	97.0	99.1	78.4	91.1	97.3	87.3	95.6	97.2
Ours	ScanNet		✓	✓	**95.5**	**97.6**	**99.1**	**80.4**	**92.2**	**97.6**	**88.9**	**96.4**	**97.6**

accuracy within three thresholds of angles, the translation accuracy within three thresholds of lengths and the Chamfer accuracy within three thresholds of metric distances, as introduced in UR&R [15]. We also include FMR [10,12] to directly compare the extracted correspondence with the reference methods, in which we use rigorous thresholds $\tau_1 = 0.05$ and $\tau_2 = 0.5$.

Baseline Methods We compare our work with the conventional registration methods, which extract 3D features by SIFT [27], SuperPoint [14] and FCGF [10], and then estimate the geometric transformation via RANSAC. Moreover, we compare with the learning-based registration approaches, such as DGR [8] and 3D MV Reg [19] as the supervised approaches, and UR&R [15], BYOC [16] as the unsupervised approaches. The results of these methods are borrowed from [15,16]. Last, we also use the RGB-D images as the input for correspondence estimation in UR&R [15] (*i.e.,* UR&R (RGB-D)), as another important baseline method.

Training Details For fair comparison, we follow the same training scheme as in [15]. Specifically, we train our model based on the 3D Match dataset for only 14 epochs with the learning rate of 1e-4. We also train our model based on the ScanNet dataset for only 1 epoch with the learning rate 1e-4. All models are trained on the machine with one NVIDIA Tesla V100 GPU. The batch size is 8. We use Adam Optimizer [25] with epsilon 1e-4 and momentum 0.9.

4.2 Experimental Results

We provide our experimental results, *i.e.,* registration errors in Table 1 and registration accuracies in Table 2. It is observed that our newly proposed method

Table 3. Comparision between our complete method (*i.e.,* the 4$^{\text{th}}$ row) and three alternative methods, which directly adopt the concatenation of RGB images and depth maps for generating the intermediate feature. "MS" means multi-scale strategy, "DC" means dilated convolutions in the visual feature extractor, and "LLT" is the local linear transformation module. All models are trained based on the 3D Match dataset.

MS	DC	LLT	Rotation					Translation					Chamfer				
			Accuracy			Error		Accuracy			Error		Accuracy			Error	
			5	10	45	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
			88.4	94.2	98.6	3.8	1.1	67.3	83.8	94.5	8.5	3.0	78.9	91.7	94.6	6.5	0.2
✓			88.5	94.4	98.6	3.8	1.1	68.1	84.5	94.8	8.3	3.0	79.5	92.1	94.9	6.3	0.2
✓	✓		90.4	95.0	98.6	3.6	1.0	70.8	86.5	95.3	8.1	2.8	81.8	93.1	95.4	6.2	0.2
✓	✓	✓	**93.4**	**96.5**	**98.8**	**3.0**	**0.9**	**76.9**	**90.2**	**96.7**	**6.4**	**2.4**	**86.4**	**95.1**	**96.8**	**5.3**	**0.1**

not only outperforms UR&R (RGB-D), but also achieves significant improvement over the baseline methods [27,14,10,16,20,19,15]. Specifically, our method trained on the 3D Match dataset achieves much better results than all other end-to-end optimized methods that are also trained on the 3D Match dataset. For example, when compared to the most recent unsupervised point cloud registration method UR&R [15], we respectively reduce 21.1% mean rotation error, 24.7% mean translation error, and 18.5% mean Chamfer distance. We also increase the FMR performance for about 11.54%, which directly validates the superior correspondence estimation performance of our method. With respect to registration accuracies, we also achieve significant gains at the strictest thresholds. These results demonstrate that our proposed network has universal registration ability, because significant gains can be achieved on the large-scale ScanNet dataset by simply training the network in a smaller 3D Match dataset.

We have similar observations when compared with these methods trained on the ScanNet dataset. But without any domain gap between training & testing data, the baseline methods can achieve almost saturated performance (over 90% in terms of most metrics for UR&R). While it is non-trivial to achieve further gains in this case, our method still reduces up to 23.7%/9.3%/12.6% relative error rate over the baselines in terms of rotation/translation/Chamfer distance.

4.3 Ablation Study and Analysis

Analysis of Each Component In Table 3, we analyse the effectiveness of each proposed component, *i.e.,* the local linear transformation (LLT) module, the multi-scale (MS) fusion strategy and dilated convolution (DC), by comparing our complete method to three alternative methods. We train all models based on the 3D Match dataset. The first variant in the 1$^{\text{st}}$ row replaces the dilation convolutions with regular convolutions, and does not adopt either multi-scale fusion strategy or the LLT module. The second variant in the 2$^{\text{nd}}$ row introduces the multi-scale fusion strategy upon the first alternative, and the third variant in the 3$^{\text{rd}}$ row further includes dilated convolution in the visual feature extractor. The first variant achieves the worst registration performance, while the second

Table 4. Comparision between our complete method and four variants, which are (1) the method without adopting the fusion mechanism at all (*i.e.,* the 3rd row in Table 3), (2) the method without the guidance map, and (3) the method that replaces LLT by the affine transformation. (4) the method that replaces LLT by multi-head cross-attention (MHCA). All models are trained on the 3D Match dataset.

Fusion Strategies	Rotation Accuracy			Rotation Error		Translation Accuracy			Translation Error		Chamfer Accuracy			Chamfer Error	
	5	10	45	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
no fusion	90.4	95.0	98.6	3.6	1.0	70.8	86.5	95.3	8.1	2.8	81.8	93.1	95.4	6.2	0.2
LLT w/o guidance map	92.3	95.9	98.8	3.2	0.9	74.7	88.7	96.3	7.0	2.5	84.6	94.4	96.2	5.4	0.1
Affine transformation	92.3	96.0	98.8	3.1	0.9	75.3	89.0	96.3	6.7	2.5	85.2	94.5	96.3	5.4	0.1
MHCA	91.3	95.1	98.4	3.8	0.9	73.5	87.6	95.2	8.4	2.6	83.4	93.3	95.4	6.5	0.2
LLT (Ours)	**93.4**	**96.5**	**98.8**	**3.0**	**0.9**	**76.9**	**90.2**	**96.7**	**6.4**	**2.4**	**86.4**	**95.1**	**96.8**	**5.3**	**0.1**

Table 5. Comparision between our method and UR&R (RGB-D) when trained based on ground truth camera poses. All models are trained on the 3D Match dataset.

Methods	Rotation Accuracy			Rotation Error		Translation Accuracy			Translation Error		Chamfer Accuracy			Chamfer Error	
	5	10	45	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
UR&R (RGB-D)	92.3	95.3	98.2	3.8	0.8	77.6	89.4	95.5	7.8	2.3	86.1	94.0	95.6	6.7	0.1
Ours	**96.5**	**97.8**	**98.8**	**2.7**	**0.8**	**83.8**	**93.8**	**97.6**	**5.8**	**2.0**	**91.2**	**96.7**	**97.6**	**4.8**	**0.1**

one reduces 2.4% mean translation error and 3.1% mean Chamfer distance when compared to the first variant. The third variant further reduces 7.6% mean rotation error, 2.4% mean translation error and 1.6% mean Chamfer distance when compared to the second alternative method. Note that our complete method in the 4th row after using the LLT module can bring the most significant gains.

Different Fusion Strategies In Table 4, we further compare our LLT based fusion module with the other alternatives. We train these models on the 3D Match dataset. The first variant in the 1st row does not adopt the fusion mechanism at all, which achieves the worst registration performance. The second one in the 2nd row directly uses the Bilateral Grid without using the guidance map, while the third variant in the 3rd row replaces the linear transformation by the affine transformation. It is observed that both the second and the third variants can intuitively bring some performance improvements. In contrast, our proposed LLT module in the 5th row can bring the most significant gains. It is interesting that the variant using affine transformation is worse than that using the linear transformation. A possible explanation is that the bias term in the affine transformation indicates another summation operation between the geometric features and the visual features, which may deteriorate the feature representation if two modalities are quite different. Last, in the 4th row, we adopt the fast multi-head cross attention (MHCA) mechanism of Linear Transformer [23]

Fig. 3. Visualization of pairwise matching results by UR&R (RGB-D) [15] and our method (trained on the 3D Match dataset [39]). We show the positive correspondence (*i.e.,* the matching error < 10cm) and the negative correspondence (*i.e.,* the matching error ≥ 10cm) as the green lines and the red lines, respectively. Best viewed on screen.

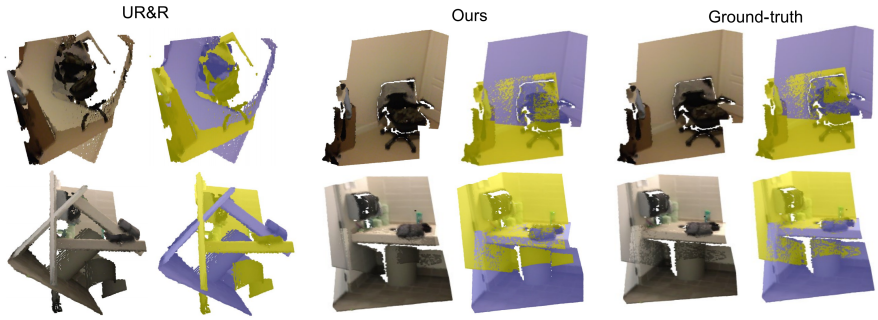

Fig. 4. Visualization of point cloud registration results from UR&R (RGB-D) [15] and our method (trained on the 3D Match dataset [39]). In the 1^{st}, the 3^{rd} and the 5^{th} columns, we show the stitched 3D scenes; while we use the purple and yellow points to represent the point clouds from the target and reference viewpoints (see the 2^{nd}, the 4^{th} and the 6^{th} columns). Best viewed on screen.

to replace the LLT module. Here, we do not apply vanilla MHCA [35] to avoid huge memory and computational costs. The results show that LLT is better than this variant in terms of all evaluation metrics.

Supervised Learning In Table 5, our proposed feature extractor can be trained under the supervised learning setting. Specifically, as in [8], we adopt the camera pose data as the ground-truth labels during the training procedure, for both our method and our baseline UR&R (RGB-D). The results show that our method is much better than UR&R (RGB-D) in terms of all metrics.

4.4 Qualitative Results

Visual Comparison about Correspondence Estimation and Registration In Figure 3, we visualize the matching results for both UR&R [15] (RGB-D) and our proposed method, in which we use the models trained on 3D Match. It is observed that our method provides more accurate matching results across two views. Taking the results in the left of Figure 3 as an example, UR&R (RGB-D) finds false correspondence around the plain area within the floor and wall,

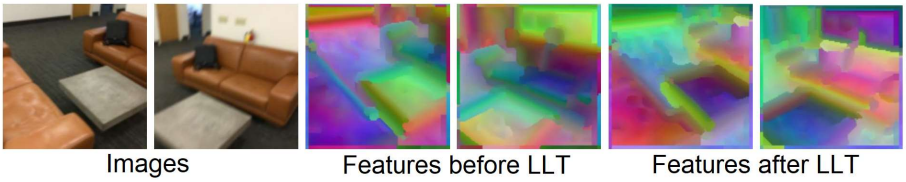

Fig. 5. The 3rd and the 4th columns represent the projected 3D features, which are the input to our LLT, while the 5th and the 6th columns represent the projected 3D features, which are the output from our LLT. We use t-SNE [28] for visualization, in which each 3D feature is mapped to the corresponding color.

while our method pays more attention to salient objects, such as the chairs, where the correspondence can be found in a more reliable and repeatable way. In Figure 4, we visualize the registration results for both UR&R [15] (RGB-D) and our method. It is observed that our method achieves better registration results. For example, in the 1st and 4th rows, our method generates very close stitching results to the ground-truth, while the results from the UR&R (RGB-D) method are completely failed. As we can already produce more accurate matching results, it is not surprised that we can achieve better point cloud registration performance than UR&R (RGB-D).

Feature Visualization In Figure 5, the 3rd and 4th columns and the 5th and 6th columns are the projected 3D features from left and right images by using t-SNE, before and after using the LLT module. We observe that the learnt features (e.g. within the table area) after using our LLT are more likely to follow the geometric structure, and have become more consistent across two views.

5 Conclusion

In this work, we have proposed a new geometric-aware visual feature extractor (GAVE) to effectively learn visual-geometric features, in which we propose multi-scale local linear transformation to progressively fuse the geometric and visual features. Our proposed GAVE module can be easily plugged into different end-to-end point cloud registration pipelines like [15] (as already discussed in this work), which significantly enhances the point cloud registration performance. Extensive experiments not only show our method outperforms the existing registration methods, but also indicate the effectiveness of our newly proposed LLT module and multi-scale fusion strategy. It is possible to further extend and apply our proposed GAVE feature extractor for more RGB-D based 3D computer vision tasks, such as recognition, tracking, reconstruction and *etc.*, which will be studied in our future work.

Acknowledgement This work was partially supported by the National Natural Science Foundation of China (No. 61906012, No. 62132001, No. 62006012).

References

1. Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y.: Spinnet: Learning a general surface descriptor for 3d point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11753–11762 (2021)
2. Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: Robust & efficient point cloud registration using pointnet. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7163–7172 (2019)
3. Bai, X., Luo, Z., Zhou, L., Chen, H., Li, L., Hu, Z., Fu, H., Tai, C.L.: Pointdsc: Robust point cloud registration using deep spatial consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15859–15869 (2021)
4. Bay, H.: Surf : Speed up robust features. In: European Conference on Computer Vision (2006)
5. Besl, P.J., Mckay, H.D.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis & Machine Intelligence **14**(2), 239–256 (1992)
6. Chen, Z., Gu, S., Lu, G., Xu, D.: Exploiting intra-slice and inter-slice redundancy for learning-based lossless volumetric image compression. IEEE Transactions on Image Processing **31**, 1697–1707 (2022)
7. Chen, Z., Lu, G., Hu, Z., Liu, S., Jiang, W., Xu, D.: Lsvc: A learning-based stereo video compression framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6073–6082 (2022)
8. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2514–2523 (2020)
9. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
10. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8958–8966 (2019)
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
12. Deng, H., Birdal, T., Ilic, S.: Ppfnet: Global context aware local features for robust 3d point matching. IEEE (2018)
13. Derpanis, K.G.: The harris corner detector. York University **2** (2004)
14. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
15. El Banani, M., Gao, L., Johnson, J.: Unsupervisedr&r: Unsupervised point cloud registration via differentiable rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7129–7139 (2021)
16. El Banani, M., Johnson, J.: Bootstrap your own correspondences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6433–6442 (2021)
17. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. ACM Transactions on Graphics (TOG) **36**(4), 1–12 (2017)

18. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. Acm Transactions on Graphics **36**(4), 118 (2017)
19. Gojcic, Z., Zhou, C., Wegner, J.D., Guibas, L.J., Birdal, T.: Learning multiview 3d point cloud registration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1759–1769 (2020)
20. Guo, Y., Wang, H., Hu, Q., Liu, H., Bennamoun, M.: Deep learning for 3d point clouds: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**(99), 1–1 (2020)
21. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: Predator: Registration of 3d point clouds with low overlap. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 4267–4276 (2021)
22. Hui, T., Ngan, K.N.: Depth enhancement using rgb-d guided filtering pp. 3832–3836 (2014)
23. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are rnns: Fast autoregressive transformers with linear attention. In: International Conference on Machine Learning. pp. 5156–5165. PMLR (2020)
24. Ke, Y.: Pca-sift : A more distinctive representation for local image descriptors. Proc. CVPR Int. Conf. on Computer Vision and Pattern Recognition, 2004 (2004)
25. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. Computer Science (2014)
26. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Transactions on Graphics (ToG) **26**(3), 96–es (2007)
27. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
28. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)
29. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
30. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
31. Rister, B., Horowitz, M.A., Rubin, D.L.: Volumetric image registration from invariant keypoints. IEEE Transactions on Image Processing **26**(10), 4900–4910 (2017)
32. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.R.: Orb: an efficient alternative to sift or surf. In: IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011 (2011)
33. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: IEEE Conference on Computer Vision & Pattern Recognition. pp. 567–576 (2015)
34. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the International Conference on Intelligent Robot Systems (IROS) (Oct 2012)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
36. Xia, X., Zhang, M., Xue, T., Sun, Z., Fang, H., Kulis, B., Chen, J.: Joint bilateral learning for real-time universal photorealistic style transfer. In: European Conference on Computer Vision. pp. 327–342. Springer (2020)

37. Xu, B., Xu, Y., Yang, X., Jia, W., Guo, Y.: Bilateral grid learning for stereo matching networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12497–12506 (2021)
38. Yu, H., Li, F., Saleh, M., Busam, B., Ilic, S.: Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. Advances in Neural Information Processing Systems **34**, 23872–23884 (2021)
39. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1802–1811 (2017)