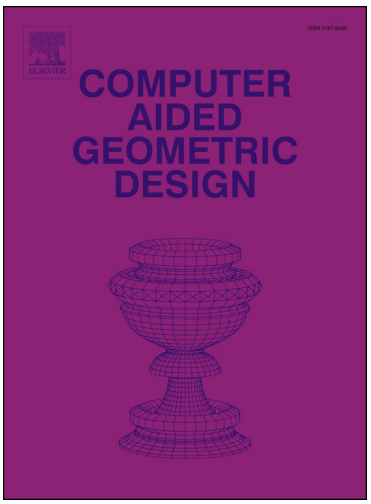



# Deep learning-based 3D local feature descriptor from Mercator projections

Masoumeh Rezaei, Mehdi Rezaeian, Vali Derhami, Ferdous Sohel, Mohammed Bennamoun

PII: S0167-8396(19)30080-9
DOI: https://doi.org/10.1016/j.cagd.2019.101771
Reference: COMAID 101771

To appear in: *Computer Aided Geometric Design*

Received date: 14 January 2019
Revised date: 12 August 2019
Accepted date: 26 August 2019

Please cite this article as: Rezaei, M., et al. Deep learning-based 3D local feature descriptor from Mercator projections. *Comput. Aided Geom. Des.* (2019), 101771, doi: https://doi.org/10.1016/j.cagd.2019.101771.

## Highlights

- This paper introduces a low complexity and low-dimensional local descriptor.
- We learn a novel data representation based on the Mercator projection.
- We achieve this using a Siamese network that directly learns from point clouds.
- We report superior performance against noise and varying mesh resolutions.

# Deep learning-based 3D local feature descriptor from Mercator projections

Masoumeh Rezaei[a], Mehdi Rezaeian[a], Vali Derhami[a], Ferdous Sohel[b,1], Mohammed Bennamoun[c]

[a] *Computer Engineering Department, Faculty of Engineering, Yazd University, Yazd, Iran*
[b] *College of Science, Health, Engineering and Education, Murdoch University, Perth, Australia*
[c] *School of Physics, Mathematics and Computing, The University of Western Australia, Perth, Australia*

**Abstract**

Point clouds provide rich geometric information about a shape and a deep neural network can be used to learn effective and robust features. In this paper, we propose a novel local feature descriptor, which employs a Siamese network to directly learn robust features from the point clouds. We use a data representation based on the Mercator projection, then we use a Siamese network to map this projection into a 32-dimensional local descriptor. To validate the proposed method, we have compared it with seven state-of-the-art descriptor methods. Experimental results show the superiority of the proposed method compared to existing methods in terms of descriptiveness and robustness against noise and varying mesh resolutions.

*Keywords:* 3D object recognition, Local descriptor, Siamese network, Deep neural network

## 1. Introduction

3D object recognition is a rapidly growing research area in computer vision. Introducing of low-cost range sensors e.g., Microsoft Kinect has made a great interest in the point cloud processing. A point cloud is a set of three-dimensional geometric points that represents the external surface of the objects. 3D object recognition methods can be classified into two broad groups: global feature-based and local feature-based methods. The global feature-based methods [1, 2, 3, 4, 5, 6, 7, 8] use a set of features for describing the geometric properties of the whole object while the local feature-based methods encode the geometric properties around a point on the 3D object surface.

The local feature-based methods are popularly used because the global feature-based methods usually need a prior segmentation of the scene and they are not robust to the clutter and occlusion [9]. Local feature-based methods consist of six main steps: keypoint detection, local feature description, feature matching, hypothesis generation, coarse registration and fine registration, and finally verification. The existing local feature descriptors can

[1] Corresponding author, f.sohel@murdoch.edu.au

be categorized in two groups, Hand-crafted feature descriptors, and Learned feature descriptors. Many Hand-crafted local descriptors have been proposed in the literature, the most popular descriptors are Splash [10], Point signature [11], Spin image [12], Finger print [13], 3D Shape Context (3DSC) [14], Unique Shape Context (USC) [15], Point Feature Histograms (PFH) [16], Fast Point Feature Histograms (FPFH) [17], Radius based Surface Descriptor (RSD) [18], Signature of Histograms of OrienTations (SHOT) [19], Rotational Projection Statistics (RoPS) [20] and Local Feature Statistics Histogram (LFSH) [21]. A comprehensive review and performance evaluation of these techniques have been presented in [22] and [23], respectively. Unfortunately, the challenges of real data, such as the presence of noise, occlusions, clutter and transformation significantly decrease performance of such descriptors.

Advances in deep convolutional neural networks have led to works for learning local descriptors. In some learned local descriptors, multi-view images of the same point are used for training the network [24, 25]. These methods cannot learn the full geometric information of the points because there are many different views for each point. A large number of images are needed to train the network that causes high complexity; therefore, they are not suitable for description. Works include PointNet [26] and PointNet++ [27], 3DMatch [28], Compact Geometric Features (CGF) [29], Point Pair Feature NET (PPFNet) [30], FoldNet [31], PPF-FoldNet [32], the method presented by Deng et. *al* [33] 3Dfeat-net [34] and 3D point-capsule networks(3D-PointCapsNet) [35] operate directly on the point clouds. 3DMatch causes high complexity because of using a 3D voxel grid as the input of the network. The network input of CGF is a histogram that describes geometric information. PPFnet uses the point pair features by incorporating global Context. PPF-foldnet is a developed combination of PointNet, FoldingNet, and PPFNet and the method presented by Deng et. *al* [33] is an extension of PPF-Foldnet. In most cases the learned descriptors outperform the hand-crafted methods but some learned 3D feature descriptors are not rotation invariant [28, 30, 34] or have high-dimensional features [28, 31, 32, 35].

In this paper, we propose a new local descriptor that directly learns from the point clouds to provide robust and precise geometric features. Our model starts with extracting geometric features using the Mercator projection of the neighborhood sphere. The Mercator projection is a cylindrical projection where the surface of a sphere is mapped into a plane preserving the angles and the shapes of small objects [36]. The Mercator map has a number of properties that make it suitable for data representations in a point cloud. These properties include **(i)** Mercator map is a conformal map projection in which any angle on Earth (a sphere or an ellipsoid) is preserved in the image of the projection; **(ii)** it maintains small element geometry, which means Mercator projection preserves the shapes of small regions. This makes the projection suitable for using on a small patch around an interest point in a point cloud; **(iii)** it can well describe the geometric properties of an interest point. In the point cloud, it can preserve these properties between any two points.

Then, we use a Siamese network [37] as a local descriptor that trains with these Mercator projections of the corresponding points where a smaller distance between two descriptors shows a higher likelihood of the similarity. The idea is that by learning from these images, the Siamese network can encode accurate and robust features that describe geometric

information around a point. We perform detailed comparisons with the state-of-the-art descriptors on the Bologna dataset [38] and 3D match dataset [28], in terms of descriptiveness and robustness against noise and varying mesh resolutions and demonstrate consistent gains in these terms.

The rest of this paper is organized as follows: Section 2 describes the proposed method, Section 3 presents the experimental results and analysis, and finally, the paper is concluded in Section 4.

## 2. The proposed method

Inspired by the success of deep learning in computer vision, we choose the Siamese networks to extract effective and robust local features. Siamese networks were introduced in 1990 by Bromley and LeCun for hand-written Signature verification. They consist of two sub-networks that share weights to find similarities or differences between two inputs [39]. The outputs are concatenated and given to a fully connected network. The goal of the Siamese networks is to learn the descriptors for comparison between the inputs and make the output features similar if input pairs are labeled as similar, and dissimilar for the input pairs which are labeled as dissimilar, in another word, they can learn both similarity and the features directly from the data. Siamese network can be easily trained using standard optimization techniques on pairs of data [40]. Another advantage of the Siamese network is that Siamese seems best suited for applications where there is only a few examples per class of data available.

We use the Siamese network to map the proposed point representation into a 32-dimensional local descriptor. The proposed method can directly learn from the point clouds, so it is more descriptive and robust.

### *2.1. Data Representation*

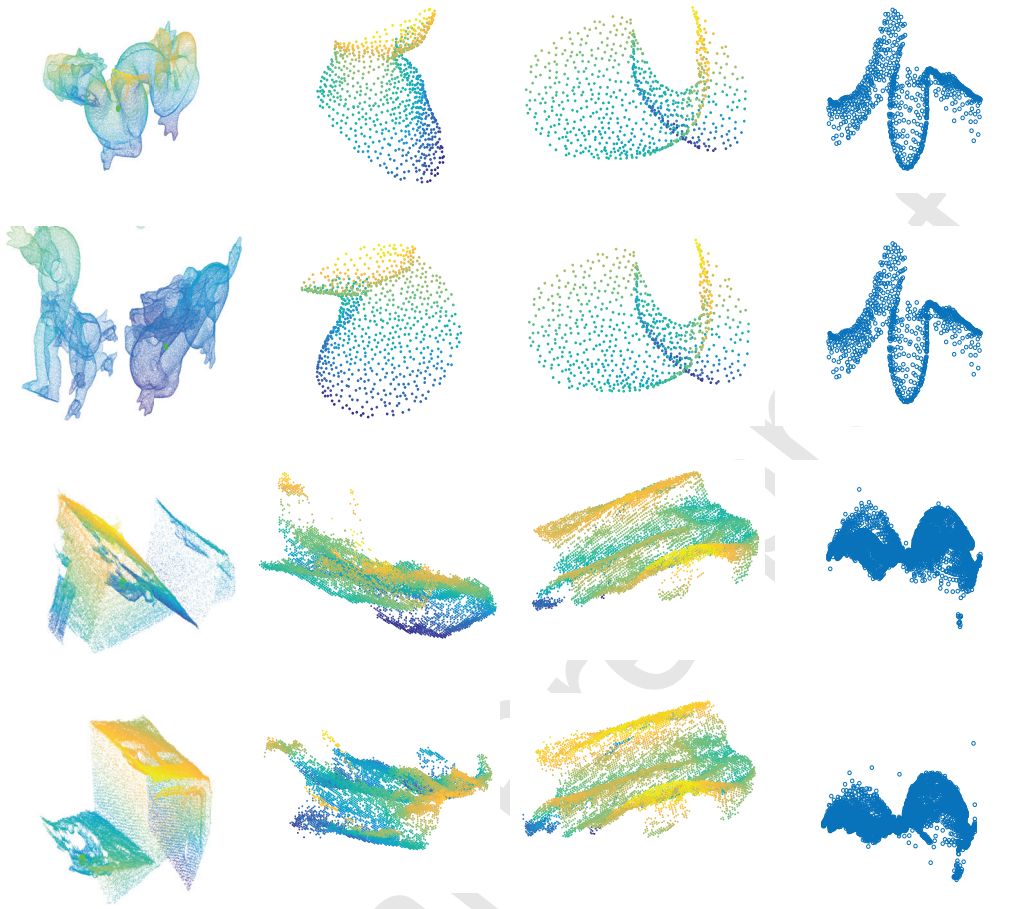
Given a query point $p$, a sphere of radius $r$ is centered at $p$ to determine the neighbors. Then Mercator projection is used for mapping the sphere into a plane with considering the Local reference frame (LRF) as previously suggested by Tombari et *al.* in 2010 [19].

The Mercator projection is a cylindrical projection that was proposed by G. Mercator in 1569. In this projection, the surface of a sphere is mapped into a plane such that the places must be positioned appropriately everywhere by considering their true distance, direction, and their relative longitude and latitude. In the point cloud, it can preserve these properties between any two points [36]. The Mercator projection for each point is identified using two following equations:

$$x = \lambda \tag{1}$$

$$y = \ln\left[\tan\left(\frac{\phi}{2} + \frac{\pi}{4}\right)\right] \tag{2}$$

where $\lambda$ is the longitude and $\phi$ is the latitude of a point in the sphere, and $(x,\ y)$ represents corresponding point in the Cartesian map. For further information about the



(a) Original point clouds (b) Neighbours in the global coordinates (c) Neighbours in the local coordinates (d) Mercator projections

Figure 1: Mercator projections of two pairs of corresponding points.

Mercator projection please see [41, 42]. Fig. 1 shows the Mercator projections of two pairs of corresponding points. In each row, first, the point cloud and the selected point are shown. The selected point is shown with a green sphere. The neighboring points are displayed in the second column. Then for better visualization, the neighboring points are shown in the local coordinates, and finally the Mercator projection of neighboring points is shown. Using the Mercator projection, there is only one representation for each point; so, the problem of multiple representations of a point or ambiguity is addressed. The first two rows show the Mercator projections of two corresponding points from Bologna dataset. As we can see from Fig. 1 the Mercator projections of these two points are similar because they describe geometric information around two points which are rigidly equivalent. The second two rows show the Mercator projections of two corresponding points selected from 3Dmatch dataset. We observe that the Mercator projections are slightly different. Sometimes it happens because of noise, occlusion, and clutter that make some parts added to the projection or removed from it. For these reasons we choose Siamese network for learning these cases.

For extracting images as the input of the Siamese network, we need ranges for achieved $x$ and $y$. The variable $x$ is in the interval $[-\pi, \pi]$ but range of $y$ is different for the Mercator projection of each keypoint. As a result, the minimum and maximum of the variable $y$ for all neighbor points are considered as the range of $y$, then a histogram $60 \times 60$ is measured. The Mercator projections of all neighbors are defined and the number of points in each bin counted. Then we normalize the histogram by dividing each bin by the total number of neighbor points, it causes more robustness to noise and mesh resolution.

### *2.2. Network architecture and training*

We define a Siamese network architecture that receives a pair of corresponding images as input. So, branches of the Siamese network can be viewed as a descriptor module and the top network as a similarity function.

The network architecture is detailed in Fig. 2, given two $60 \times 60$ images, the output is the probability that the two input images match. The architecture of each subnetwork is as follows: Convolution layer ($8 \times 2 \times 2$)- ReLU- Convolution layer ($16 \times 2 \times 2$)- ReLU-Max pooling ($2 \times 2$)- Convolution layer ($16 \times 2 \times 2$)- ReLU- Convolution layer ($32 \times 2 \times 2$)- ReLU-Max pooling ($2 \times 2$)-Flatten (4608)-Dense (32)-Dropout. The output of each subnetwork is a 32-dimensional feature which is considered as a local descriptor. The second part of the network is defined as a similarity function that is as follows: Concatenate layer (for merging two sublayer outputs)-Dense (16)-ReLU- Dense (4)-ReLU-Dense (1). The output of this part of the network is the probability of images similarity. Note that linear activation function is used for all convolutional and dense layers except the last dense layer that uses Sigmoid as the activation function. Mean absolute error and binary cross entropy are used as metric function and loss function, respectively.

## 3. Experimental results and discussion

In this section, we evaluated the performance of the proposed method on the Bologna [38] and 3D match datasets [28]. The Bologna dataset consists of six models namely Armadillo, Dragon, Stanford Bunny, Happy Buddha, Chinese Dragon and Thai Statue, and 45 scenes that each scene is generated by three to five rigidly transformed models. 3Dmatch dataset is an ensemble of SUN3D [43], 7-Scenes [44], RGB-D Scenes v.2 [45], BundleFusion [46] and Analysis-by-Synthesis [47] datasets.

### *3.1. Bologna dataset*

The dataset contains 45 scenes, 25 scenes are used as train set and 20 scenes as the test set. To obtain the corresponding images, 5000 points are randomly selected from train set. Then, their corresponding scene keypoints are determined and Mercator projections are extracted for all of these corresponding points; so, 5000 pairs with 100% image similarity are extracted. The pairs are split randomly into 80% train and 20% test for learning the networks. As each sublayer figures a local descriptor, the red block is used for testing the features. The network was trained for 10 epochs using a batch size of 32. The number of steps per epoch was 500. Training took about 3 hours. Fig. 3 shows the history of loss and

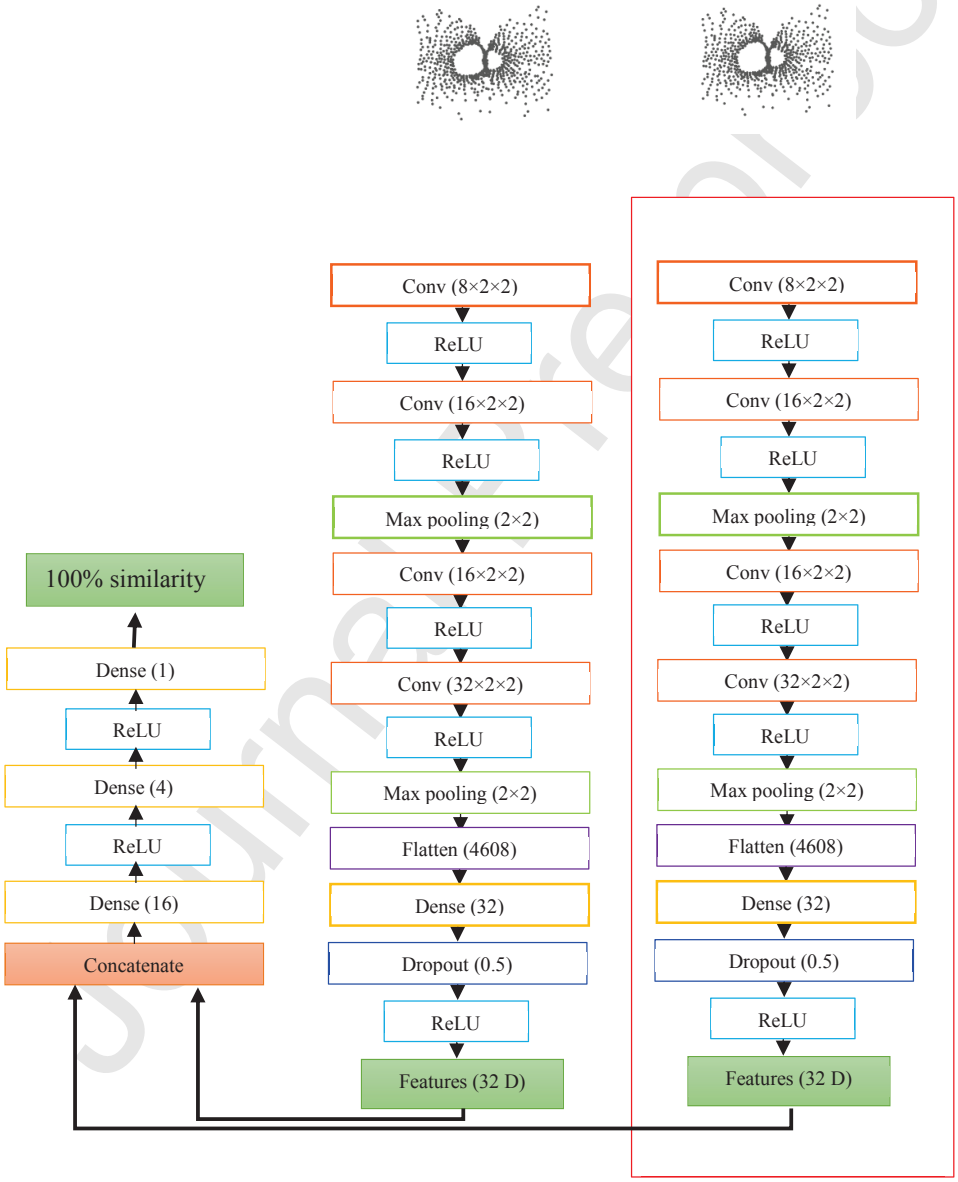



Figure 2: The proposed Siamese network architecture

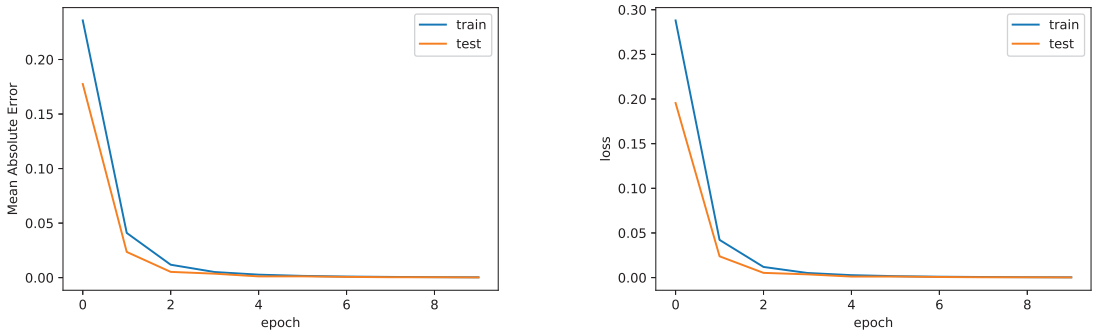



Figure 3: Loss and error plots on the Bologna dataset.

accuracy in the learning process. As can be seen, the model converged quickly and achieved highly accurate results.

In this section, we compared the proposed method with several state-of-the-art hand-crafted approaches including LFSH, 3DSC, PFH, FPFH, SHOT, RoPS, and RSD in terms of descriptiveness and robustness against noise and varying mesh resolutions using the Bologna dataset. All of the experiments were conducted on a computer with Intel (R) Core (TM) i5-45705 CPU 2.90 GHz, 8GB RAM memory. We used the publicly available MATLAB implementation of the RoPS method[2]. We implemented the LFSH method using the source C++ code. The other methods are publicly available in the Point Cloud Library (PCL)[3]. The proposed method was implemented in the TensorFlow framework [48].

*3.1.1. Comparison of results in term of RPC*

We applied the Recall versus 1-Precision Curve (RPC) to compare the descriptiveness of the proposed method with the mentioned descriptors. The default parameters in the PCL implementations were applied for all selected descriptors. At first, 500 key points were randomly selected for all models in each scene; then, the corresponding scene keypoints were determined. It should be noted that the selected keypoints and the support radius were the same for all methods for fairness, the support radius was set as 15 mr (mesh resolution). Then, the RPC was evaluated.

As previously reported in [38] the RPC is generated as follows: for each scene feature, the ratio between the closest and the second closest features is calculated. The scene feature and its closest model feature are considered as a match if this ratio is smaller than a threshold $\tau$. A match is considered as a correct one if the distance between the scene keypoint and the ground-truth transformed model keypoint is sufficiently small. The precision and the recall are the number of correct matches divided by the total number of matches and the total number of correspondences, respectively. The PRC curve is generated by changing the threshold $\tau$.

[2]http://yulanguo.me/
[3]www.pointclouds.org

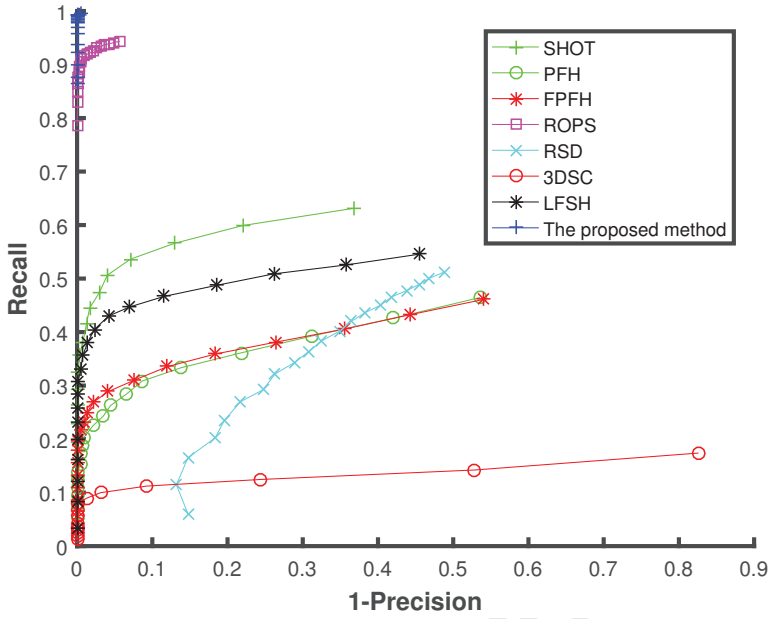



Figure 4: The RPC-based evaluation of the methods.

Fig. 4 shows the RPC results on the noise-free Bologna dataset. The threshold for finding true positive matches was set as 15 mr. The results are the average of precision and recall for all scenes in the test set. The proposed method has attained excellent result followed by RoPS, SHOT, LFSH, FPFH, PFH, RSD, and 3DSC. The RoPS achieved good results, SHOT and LFSH acted reasonably but the results of other methods are not good. As expected, PFH and FPFH had a similar performance. Note that, the highest recall value for the proposed method is 0.99. These results demonstrate that our method effectively describes geometric information around a point.

*3.1.2. Robustness to noise*

We compared the proposed method with the mentioned descriptors in term of robustness in the different level of noises. We added Gaussian noise with the standard deviations 0.1, 0.3 and 0.5 mr to the scene data. The RPC results in the different levels of Gaussian noise are shown in Fig. 5. As can be seen, it is clear that our proposed descriptor achieved the best results in noise with standard deviation 0.1 mr. It is highly robust to this level of noise. The highest recall rate is achieved by the proposed method, which is 0.93 while the second best is by RoPS, which is 0.69. This underscores the robustness of the proposed method with respect to noise.

Also with noise 0.3 mr, the proposed method has significantly outperformed other methods. Although the SHOT descriptor provides the best performance under high-level noise for the small thresholds, the highest recalls for the proposed method and the SHOT method are 0.48 and 0.46, respectively which shows the superiority of our method at this level of noise.

The superior performance of the proposed method is due to the fact that noise causes some changes in the Mercator projection but the main behavior of local neighbors is preserved unless the geometric structure is completely changed with noise. The Siamese network can handle noise because it trains with images that are not completely similar but with the same main structure.

*3.1.3. Robustness to varying mesh resolution*

In order to evaluate the robustness of the methods to varying mesh resolution, we simplified the scene meshes with different reduction factors such as 1/8, 1/4 and 1/2. RPC results under different reduction factors are presented in Fig. 6. As depicted, the proposed method outperformed the other methods in the simplification with reduction factors 1/8 and 1/4 (Fig. 6a and Fig. 6b respectively). At the high level of reduction factor, i.e. 1/2, the RoPS outperformed the other methods and the proposed method was the second best method. The RSD was very sensitive to varying mesh resolutions but the performance of other methods dropped by some margin. The highest recall for the proposed method with reduction factors 1/8, 1/4 and 1/2 are 0.81, 0.61 and 0.35, respectively.

These observations can be explained based on the following two reasons. First, mesh decimation preserves the main structure of the Mercator projection which makes the proposed method efficient. Second, The Siamese network inputs are derived by a normalized histogram which causes more robustness.

*3.1.4. Pairwise Registration Results*

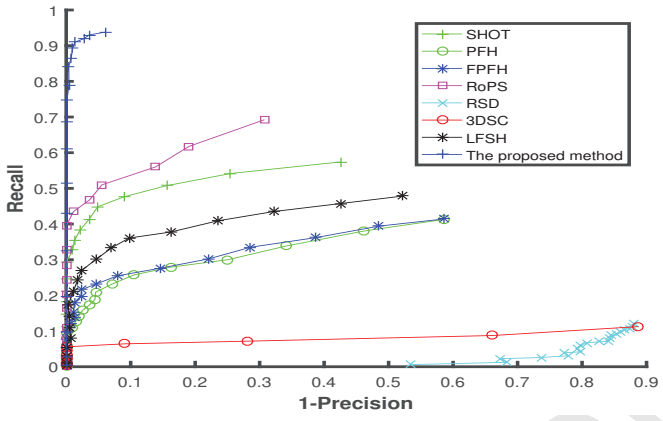
We created 4 random transformation matrices for each model in the Bologna dataset to generate 24 pairs of model and scene. At the beginning for each pair, we randomly selected 100 points for the model and 100 points for the scene. Then, features were extracted as described before. The RANSAC method was applied for the coarse registration. Fig. 7 shows the coarse registration results of the proposed method for one of the pairs. As can be observed, the transformed model is completely matched with the scene. We measured the root mean square error (RMSE), rotation error $\theta_r$, translation error $d_t$ and correct registration rate to compare with the other methods. The RMSE, $\theta_r$ and $d_t$ are calculated as follow [49]:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N} \parallel R_e\ p_k + t_e - q_k \parallel^2}{N}} \tag{3}$$

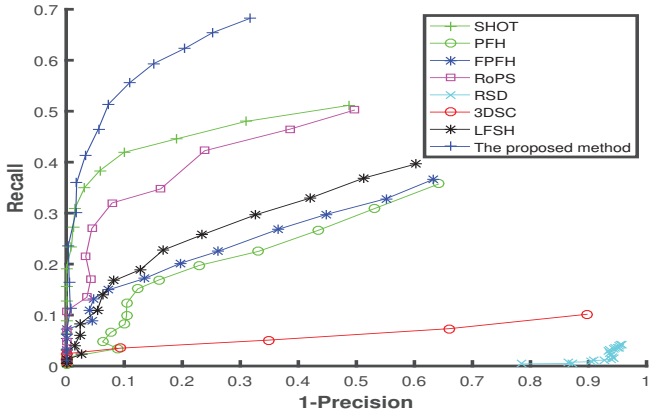
$$\theta_r = \arccos \frac{trace(R_{GT}\ {R_e}^T) - 1}{2}\ \frac{180}{\pi} \tag{4}$$

$$d_t = \parallel T_{GT} - T_e \parallel \tag{5}$$

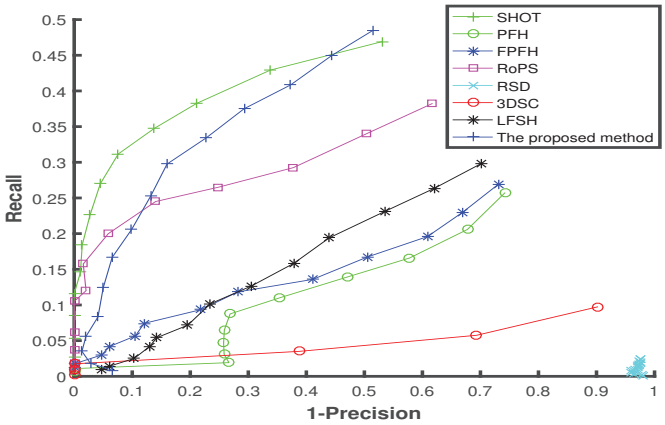
where $p_k$ and $q_k$ are two rigidly equivalent points, $N$ is the number of points, $R_{GT}$ and $R_e$ are ground truth and estimated rotation matrices, respectively. $T_{GT}$ is ground truth and $T_e$ is estimated translation vectors. A pairwise registration is measured as correct if $\theta_r < 5^\circ$ and $d_t < 5 * mr$. The correct registration rate is the number of correct registrations divided by the total number of pairs. As listed in Table. 1, the RoPS and the proposed method attained




(a)

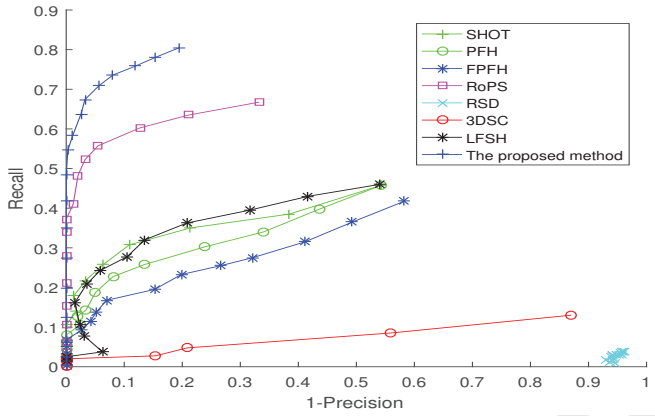



(b)

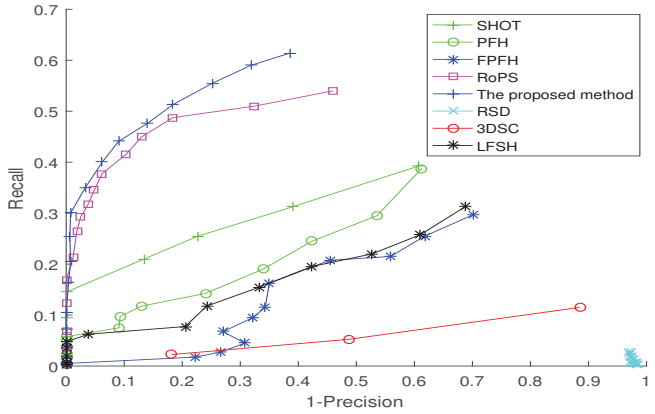



(c)

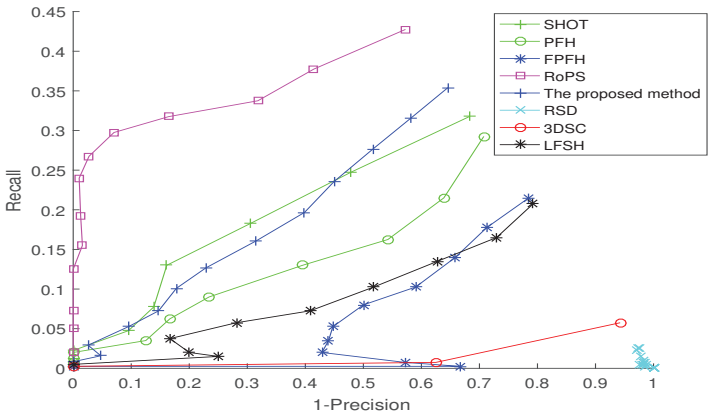
Figure 5: RPC based evaluation of the descriptors with respect to different levels of Gaussian noise (a) noise deviation 0.1 mr (b) noise deviation 0.3 mr. (c) noise deviation 0.5 mr.




(a)




(b)




(c)

Figure 6: RPC based evaluation of the descriptors with respect to different mesh resolutions (a) reduction factor 1/8 (b) reduction factor 1/4 (c) reduction factor 1/2.

Figure 7: Pairwise registration result of the proposed method.

Table 1: Pairwise registration results on the Bologna dataset

| The methods | RMSE | Translation error | Rotation error | Correct registration |
|---|---|---|---|---|
| The proposed method | **0.0000** | **0.0000** | **0.0218** | **1** |
| FPFH[17] | 2.2498 | 2.2502 | 92.0649 | 0.2500 |
| RoPS[20] | **0.0000** | **0.0000** | 0.0279 | **1** |
| RSD[18] | 0.8852 | 0.8756 | 97.1618 | 0.0417 |
| 3DSC[14] | 1.6754 | 1.6756 | 96.5474 | 0 |
| LFSH[21] | 1.7327 | 1.7208 | 77.3431 | 0.1667 |
| PFH[16] | 1.2975 | 1.2992 | 98.8903 | 0.2500 |
| SHOT[19] | 0.6092 | 0.6169 | 44.2603 | 0.6667 |

the best registration performance with 100% correct registration rate, the rotation error of our method was better than the RoPS method. The SHOT method achieved 66% correction registration rate but the performance of the other methods were rather low. The superior performance of the proposed method is due to we use the Mercator projection as the data representation. In the point cloud, this projection can preserve true distance, direction, and their relative longitude and latitude between any two neighbor points regardless of rotation and translation.

### *3.2. 3Dmatch dataset*

For evaluation of the proposed method in a real data set, in the presence of noise and occlusion, we used the 3DMatch Dataset [28]. The proposed method is compared with some learned descriptors including 3DMatch [28], CGF [29], PPFNet [30], FoldNet [31], PPF-

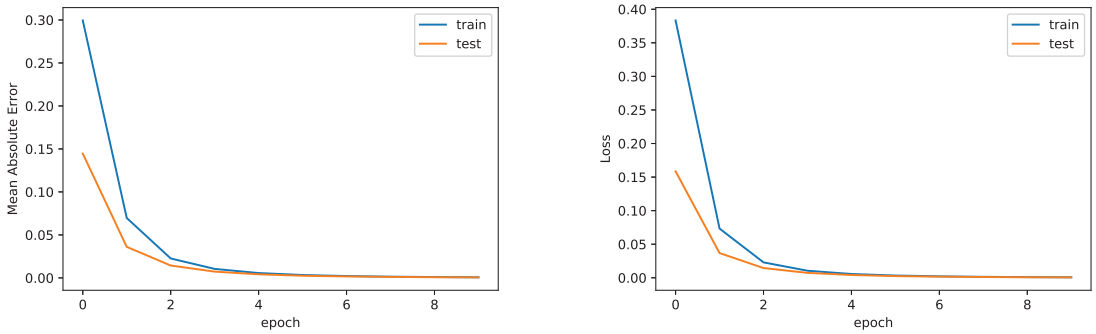



Figure 8: Loss and error plots on the 3DMatch dataset.

FoldNet [32], the method proposed by Deng et. *al* [33] and 3D-PointCapsNet [35]. The experiments were conducted on a computer with Intel (R) Core (TM) i7-8700K CPU 3.70 GHz × 12, with GeForce GTX 1080 Ti/PCIe/SSE2 GPU card.

First, the performance of the achieved features is measured by the fragment matching recall, then we evaluate our method qualitatively. The dataset contains 62 scenes, the train part contains 54 sets consist of 2D RGB-D patches and the test part contains eight scenes split into several partially overlapping fragments. Each fragment is a 3D point cloud of a surface, integrated from 50 depth frames. We used 3Dmatch toolbox for constructing fragments of the training sets. 120K corresponding points are extracted and split randomly into 80% train and 20% test and the network was trained for 10 epochs. The number of steps per epoch was 500.Training took about 30 minutes. As training proceeds, the history of loss and error are shown in Fig. 8. We observe from this figure that the model converged quickly to accurate results without overfitting.

*3.2.1. Comparison of results in term of fragment matching recall*

The fragment matching recall is generated as follows [32]: assuming a pair of fragments $P = \{p_i \in R_3\}$ and $Q = \{q_i \in R_3\}$ are aligned by the rigid transformation $T_{GT}$, the inter point pair set $M$ is calculated as follow:

$$M = \{\{p_i, q_i\}, g(p_i) = NN(g(q_i), g(P)), g(q_i) = NN(g(p_i), g(Q))\} \tag{6}$$

Where $NN$ means nearest neighbor search and $g(p)$ and $g(P)$ mean the feature for point $p$ and the pool of features for the points in $P$. Then true matches set $M_{GT}$, inlier ratio $r_{in}$, and fragment matching recall $R$ are defined as follow:

$$M_{GT} = \{\{p_i, q_i\}, (p_i, q_i) \in M, \|p_i - T_{GT}\ q_i\|_2 < \tau_1\} \tag{7}$$

$$r_{in} = \frac{|M_{GT}|}{|M|} \tag{8}$$

$$R = \frac{1}{|S|} \sum_{i=1}^{|S|} number\ of\ (r_{in}(S_i = (P_i, Q_i)) > \tau_2) \tag{9}$$

Table 2: Fragment matching recall results on the 3DMatch benchmark.

| | 3DMatch [28] | CGF [29] | PPFNet [30] | FoldNet [31] | PPF-FoldNet [32] | [33] | 3D-PointCapsNet [35] | ours |
|---|---|---|---|---|---|---|---|---|
| Kitchen | 0.5751 | 0.4605 | 0.8972 | 0.5949 | 0.7352 | 0.7964 | 0.8518 | **0.9243** |
| Home 1 | 0.7372 | 0.6154 | 0.5577 | 0.7179 | 0.7564 | 0.8077 | 0.8333 | **0.9057** |
| Home 2 | 0.7067 | 0.5625 | 0.5913 | 0.6058 | 0.6250 | 0.6971 | 0.7740 | **0.9119** |
| Hotel 1 | 0.5708 | 0.4469 | 0.5796 | 0.6549 | 0.6593 | 0.7257 | 0.7699 | **0.8901** |
| Hotel 2 | 0.4423 | 0.3846 | 0.5769 | 0.4231 | 0.6058 | 0.6731 | 0.7308 | **0.8718** |
| Hotel 3 | 0.6296 | 0.5926 | 0.6111 | 0.6111 | 0.8889 | 0.9444 | 0.9444 | **0.9630** |
| Study | 0.5616 | 0.4075 | 0.5342 | 0.7123 | 0.5753 | 0.6986 | 0.7397 | **0.7692** |
| MIT lab | 0.5455 | 0.3506 | 0.6364 | 0.5844 | 0.5974 | 0.6234 | 0.6494 | **0.8444** |
| Average | 0.5961 | 0.4776 | 0.6231 | 0.6130 | 0.6804 | 0.6234 | 0.7867 | **0.8850** |

where $|S|$ is the number of ground-truth matching non-consecutive fragment pairs, having at least 30% overlap with each other under ground-truth transformation.

The 3Dmatch test set consists of 8 scene sets, Red Kitchen data is from the 7-scenes [44] dataset and the rest are from SUN3D [43] dataset. Similar to previous research, we used 2K points per fragment and the radius is 0.3m. $\tau_1$ and $\tau_2$ are set to 0.1m and 0.05, respectively. The fragment matching recalls are reported in Table. 2. Note that the results of [33] and 3D-PointCapsNet method are reported from the original papers and the results of other methods are reporeted from PPF-Foldnet paper [32].

As demonstrated in the Table. 2, our method outperforms other methods in all scene sets. The proposed method has achieved the best results followed by 3D-PointCapsNet, PPF-FoldNet, [33], PPFNet, FoldNet, 3Dmatch and CGF methods. The method outperforms the 3D-PointCapsNet as the second best method by the margin 9.83% on average. Our method not only outperforms the other methods but also reduce the complexity of learning by projecting 3D to 2D using the Mercator projection. Another benefit is achieving low-dimensional features that causes lower data storage, complexity and error and higher speed. The number of feature dimensions in 3Dmatch, CGF, Foldnet, and PPF-FoldNet is 512. The features of PPFNet are 64-dimensional. Just The CGF has 32-dimensional features like ours. Also in comparison with the hand-crafted methods, except LFSH that has 30-dimensional features, the proposed method has the lowest number of dimensions.

We also evaluated our method qualitativly, Random Sample Consensus (RANSAC) [50] is used for coarse registration. For each scene set, one fragment pair is selected and the registration results of our method are displayed in Fig. 9. In each row, from left to right, the model, scene, our result and ground-truth result are shown. From Fig. 9, it can be seen that the proposed method can achieve the registration results close to ground-truth. The qualitative and quantitative results show that the proposed method can achieve good performance on unseen real data in the presence of noise and occlusion.

## 4. Conclusion

In this paper, we introduced a low complexity and low-dimensional local descriptor based on the Siamese network that directly learns from the point clouds. We presented a novel

(a) Models (b) Scenes (c) Our results (d) Ground-truth

Figure 9: Qualitative results of 3Dmatch dataset.

data representation based on the Mercator projection. Using the Mercator projection not only causes encoding rich and precise geometric information but also the problem of multiple representations of a point or ambiguity is addressed because there is only one representation for each point. Using this projection, the input of the network is 2D so the architecture overcame many challenges brought in by 3D data. We also demonstrated the superiority of the proposed approach to the state-of-the-art methods in terms of descriptiveness and robustness against noise and varying mesh resolutions on the Bologna and 3Dmatch dataset.

## References


[1] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, G. Bradski, Cad-model recognition and 6dof pose estimation using 3d cues, in: 2011 IEEE international conference on computer vision workshops (ICCV workshops), IEEE, 2011, pp. 585–592.

[2] Z.-C. Marton, D. Pangercic, N. Blodow, M. Beetz, Combined 2d–3d categorization and classification for multimodal perception systems, The International Journal of Robotics Research 30 (11) (2011) 1378–1402.

[3] R. B. Rusu, G. Bradski, R. Thibaux, J. Hsu, Fast 3d recognition and pose using the viewpoint feature histogram, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 2155–2162.

[4] W. Wohlkinger, M. Vincze, Ensemble of shape functions for 3d object classification, in: 2011 IEEE international conference on robotics and biomimetics, IEEE, 2011, pp. 2987–2992.

[5] D. Fehr, W. J. Beksi, D. Zermas, N. Papanikolopoulos, Covariance based point cloud descriptors for object detection and recognition, Computer Vision and Image Understanding 142 (2016) 80–93.

[6] S. H. Kasaei, A. M. Tomé, L. S. Lopes, M. Oliveira, Good: A global orthographic object descriptor for 3d object recognition and manipulation, Pattern Recognition Letters 83 (2016) 312–320.

[7] J. C. Rangel, J. Martínez-Gómez, C. Romero-González, I. García-Varea, M. Cazorla, Semi-supervised 3d object recognition through cnn labeling, Applied Soft Computing 65 (2018) 603–613.

[8] S. Bu, L. Wang, P. Han, Z. Liu, K. Li, 3d shape recognition and retrieval based on multi-modality deep learning, Neurocomputing 259 (2017) 183–193.

[9] N. Bayramoglu, A. A. Alatan, Shape index sift: Range image recognition using local features, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 352–355.

[10] F. Stein, G. Medioni, Structural indexing: Efficient 3-d object recognition, IEEE Transactions on Pattern Analysis & Machine Intelligence (2) (1992) 125–145.

[11] C. S. Chua, R. Jarvis, Point signatures: A new representation for 3d object recognition, International Journal of Computer Vision 25 (1) (1997) 63–85.

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

[13] Y. Sun, M. A. Abidi, Surface matching by 3d point's fingerprint, in: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vol. 2, IEEE, 2001, pp. 263–269.

[14] A. Frome, D. Huber, R. Kolluri, T. Bülow, J. Malik, Recognizing objects in range data using regional point descriptors, in: European conference on computer vision, Springer, 2004, pp. 224–237.

[15] F. Tombari, S. Salti, L. Di Stefano, Unique shape context for 3d data description, in: Proceedings of the ACM workshop on 3D object retrieval, ACM, 2010, pp. 57–62.

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

[17] R. B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (fpfh) for 3d registration, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 3212–3217.

[18] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, M. Beetz, General 3d modelling of novel objects from a single view, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 3700–3705.

[19] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: European conference on computer vision, Springer, 2010, pp. 356–369.

[20] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, J. Wan, Rotational projection statistics for 3d local surface description and object recognition, International journal of computer vision 105 (1) (2013) 63–86.

[21] J. Yang, Z. Cao, Q. Zhang, A fast and robust local descriptor for 3d point cloud registration, Information Sciences 346 (2016) 163–179.

[22] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, 3d object recognition in cluttered scenes with local surface features: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (11) (2014) 2270–2287.

[23] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, N. M. Kwok, A comprehensive performance evaluation of 3d local feature descriptors, International Journal of Computer Vision 116 (1) (2016) 66–89.

[24] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, F. Moreno-Noguer, Discriminative learning of deep convolutional feature point descriptors, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 118–126.

[25] K. M. Yi, E. Trulls, V. Lepetit, P. Fua, Lift: Learned invariant feature transform, in: European Conference on Computer Vision, Springer, 2016, pp. 467–483.

[26] C. R. Qi, H. Su, K. Mo, 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, 2017, pp. 652–660.

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

[28] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, T. Funkhouser, 3dmatch: Learning local geometric descriptors from rgb-d reconstructions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1802–1811.

[29] M. Khoury, Q.-Y. Zhou, V. Koltun, Learning compact geometric features, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 153–161.

[30] H. Deng, T. Birdal, S. Ilic, Ppfnet: Global context aware local features for robust 3d point matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 195–205.

[31] Y. Yang, C. Feng, Y. Shen, D. Tian, Foldingnet: Point cloud auto-encoder via deep grid deformation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 206–215.

[32] H. Deng, T. Birdal, S. Ilic, Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 602–618.

[33] H. Deng, T. Birdal, S. Ilic, 3d local features for direct pairwise registration, arXiv preprint arXiv:1904.04281.

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

[35] Y. Zhao, T. Birdal, H. Deng, F. Tombari, 3d point-capsule networks, arXiv preprint arXiv:1812.10775.

[36] E. L. Eisenstein, The printing revolution in early modern Europe, Cambridge University Press, 2005.

[37] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a "siamese" time delay neural network, in: Advances in neural information processing systems, 1994, pp. 737–744.

[38] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: European conference on computer vision, Springer, 2010, pp. 356–369.

[39] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a "siamese" time delay neural network, in: Advances in neural information processing systems, 1994, pp. 737–744.

[40] G. Koch, R. Zemel, R. Salakhutdinov, Siamese neural networks for one-shot image recognition, 2015.

[41] D. Salomon, Transformations and projections in computer graphics, Springer Science & Business Media, 2007.

[42] M. Denny, The science of navigation: from dead reckoning to GPS, JHU Press, 2012.

[43] J. Xiao, A. Owens, A. Torralba, Sun3d: A database of big spaces reconstructed using sfm and object labels, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1625–1632.

[44] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, A. Fitzgibbon, Scene coordinate regression forests for camera relocalization in rgb-d images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2930–2937.

[45] K. Lai, L. Bo, D. Fox, Unsupervised feature learning for 3d scene labeling, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 3050–3057.

[46] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, C. Theobalt, Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration, ACM Transactions on Graphics (ToG) 36 (4) (2017) 76a.

[47] J. Valentin, A. Dai, M. Nießner, P. Kohli, P. Torr, S. Izadi, C. Keskin, Learning to navigate the energy landscape, in: 2016 Fourth International Conference on 3D Vision (3DV), IEEE, 2016, pp. 323–332.

[48] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.

[49] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, M. Lu, An accurate and robust range image registration algorithm for 3d object modeling, IEEE Transactions on Multimedia 16 (5) (2014) 1377–1390.

[50] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.