Learning and Matching Multi-View Descriptors for Registration of Point Clouds

Lei Zhou[1†][0000−0003−4988−5084], Siyu Zhu[1‡][0000−0003−0293−0044],
Zixin Luo[1†][0000−0001−6946−2826], Tianwei Shen[1†][0000−0002−3290−2258],
Runze Zhang[1†*][0000−0001−9698−0178], Mingmin Zhen[1][0000−0002−8180−1023],
Tian Fang[2][0000−0002−5871−3455], and Long Quan[1][0000−0001−8148−1771]

[1] Hong Kong University of Science and Technology
{lzhouai,szhu,zluoag,tshenaa,rzhangaj,mzhen,quan}@cse.ust.hk
[2] Shenzhen Zhuke Innovation Technology (Altizure)
fangtian@altizure.com

Abstract. Critical to the registration of point clouds is the establishment of a set of accurate correspondences between points in 3D space. The correspondence problem is generally addressed by the design of discriminative 3D local descriptors on the one hand, and the development of robust matching strategies on the other hand. In this work, we first propose a multi-view local descriptor, which is learned from the images of multiple views, for the description of 3D keypoints. Then, we develop a robust matching approach, aiming at rejecting outlier matches based on the efficient inference via belief propagation on the defined graphical model. We have demonstrated the boost of our approaches to registration on the public scanning and multi-view stereo datasets. The superior performance has been verified by the intensive comparisons against a variety of descriptors and matching methods.

Keywords: Point cloud registration · 3D descriptor · Robust matching

1 Introduction

Registration of point clouds integrates 3D data from different sources into a common coordinate system, serving as an essential component of many high-level applications like 3D modeling [1, 2], SLAM [3] and robotic perception [4]. Critical to a registration task is the determination of correspondences between spatially localized 3D points within each cloud. To tackle the correspondence problem, on the one hand, a bunch of 3D local descriptors [5–12] have been developed to facilitate the description of 3D keypoints. On the other hand, matching strategies [13–15] have also been progressing towards higher accuracy and robustness.

[†] Lei Zhou and Zixin Luo were summer interns, and Tianwei Shen and Runze Zhang were interns at Everest Innovation Technology (Altizure).

[‡] Siyu Zhu is with Alibaba A.I. Labs since Oct. 2017.

[*] Runze Zhang is the corresponding author.

The exploration of 3D geometric descriptors has long been the focus of interest in point cloud registration. It involves the hand-crafted geometric descriptors [5–9] as well as the learned ones [10–12, 16]. Both kinds of methods mainly rely on 3D local geometries. Meanwhile, with the significant progress of CNN-based 2D patch descriptors [17–21], more importance is attached to leveraging the 2D projections for the description of underlying 3D structures [22–25]. Particularly for the point cloud data generally co-registered with camera images [26–29], the fusion of multiple image views, which has reported success on various tasks [30–33], is expected to further improve the discriminative power of 3D local descriptors. With this motivation, we propose a multi-view descriptor, named MVDesc, for the description of 3D keypoints based on the synergy of the multi-view fusion techniques and patch descriptor learning. Rather than a replacement, the MVDesc is well complementary to existing geometric descriptors [5–12].

Given the local descriptors, the matching problem is another vital issue in point cloud registration. A set of outlier-free point matches is desired by most registration algorithms [15, 34–38]. Currently, the matching strategies, *e.g.*, the nearest neighbor search, mutual best [15] and ratio test [13], basically estimate correspondences according to the similarities of local descriptors alone. Without considering the global geometric consistency, these methods are prone to spurious matches between locally-similar 3D structures. Efforts are also spent on jointly solving the outlier suppression via line process and the optimization of global registration [15]. But the spatial organizations of 3D point matches are still overlooked when identifying outliers. To address this, we develop a robust matching approach by explicitly considering the spatial consistency of point matches in 3D space. We seek to filter outlier matches based on a graphical model describing their spatial properties and provide an efficient solution via belief propagation.

The main contributions of this work can be summarized twofold. 1) We are the first to leverage the fusion of multiple image views for the description of 3D keypoints when tackling point cloud registration. 2) The proposed effective and efficient outlier filter, which is based on a graphical model and solved by belief propagation, remarkably enhances the robustness of 3D point matching.

2 Related Works

3D local descriptor. The representation of a 3D local structure used to rely on traditional geometric descriptors such as Spin Images [5], PFH [6], FPFH [7], SHOT [8], USC [9] and *et al.*, which are mainly produced based on the histograms over local geometric attributes. Recent studies seek to learn descriptors from different representations of local geometries, like volumetric representations of 3D patches [10], point sets [12] and depth maps [16]. The CGF [11] still leverages the traditional spherical histograms to capture the local geometry but learns to map the high-dimensional histograms to a low-dimensional space for compactness.

Rather than only using geometric properties, some existing works refer to extracting descriptors from RGB images that are commonly co-registered with point clouds as in scanning datasets [26–28] and 3D reconstruction datasets [29, 39]. Registration frameworks like [22–25] use SIFT descriptors [13] as the representations of 3D keypoints based on their projections in single-view RGB images. Besides, the other state-of-the-art 2D descriptors like DeepDesc [17], L2-Net [18] and *et al.* [19–21] can easily migrate here for the description of 3D local structures.

Multi-view fusion. The multi-view fusion technique is used to integrate information from multiple views into a single representation. It has been widely proved by the literature that the technique effectively boosts the performance of instance-level detection [30], recognition [31, 32] and classification [33] compared with a single view. Su *et al.* [30] first propose a probabilistic representation of a 3D-object class model for the scenario where an object is positioned at the center of a dense viewing sphere. A more general strategy of multi-view fusion is *view pooling* [31–33, 40], which aggregates the feature maps of multiple views via element-wise maximum operation.

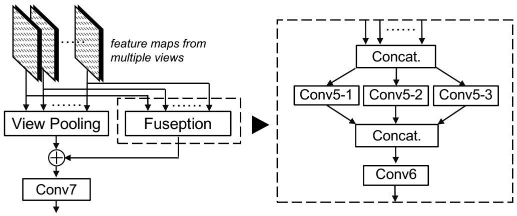
Matching. The goal of matching for registration is to find correspondences across 3D point sets given keypoint descriptors. Almost all the registration algorithms [15, 34–38] demand accurate point correspondences as input. Nearest-neighbor search, mutual best filtering [15] and ratio test [13] are effective ways of searching for potential matches based on local similarities for general matching tasks. However, as mentioned above, these strategies are prone to mismatches without considering the geometric consistency. To absorb geometric information, [41] and [42] discover matches in geometric agreement using game-theoretic scheme. Ma *et al.* [43] propose to reject outliers by enforcing consistency in local neighborhood. Zhou *et al.* [15] use a RANSAC-style tuple test to eliminate matches with inconsistent scales. Besides, the line process model [44] is applied in registration domain to account for the presence of outliers implicitly [15].

3 Multi-View Local Descriptor (MVDesc)

In this section, we propose to learn multi-view descriptors (MVDesc) for 3D keypoints which combine multi-view fusion techniques [30–33] with patch descriptor learning [17–21]. Specifically, we first propose a new view-fusion architecture to integrate feature maps across views into a single representation. Second, we build the MVDesc network for learning by putting the fusion architecture above multiple feature networks [45]. Each feature network is used to extract feature maps from the local patch of each view.

3.1 Multi-View Fusion

Currently, *view pooling* is the dominant fusion technique used to merge feature maps from different views [31–33, 40]. However, as reported by the literature [32, 46, 47], the pooling operation is somewhat risky in terms of feature aggregation due to its effect of smoothing out the subtle local patterns. Inspired by ResNet [48], we propose an architecture termed *Fuseption-ResNet* which uses the view



(a) Fuseption-ResNet

(b) Fuseption

Layers	Kernel	Channels	Stride	Activation
Conv5-1	3×3	32	1	ReLU
Conv5-2	1×3	32	1	ReLU
Conv5-3	3×1	32	1	ReLU
Conv6	1×1	-	1	ReLU
Conv7	3×3	-	2	-

(c) Parameters

Fig. 1: An overview of proposed Fuseption-ResNet (FRN). (a) Architecture of FRN that fuses feature maps of multiple views. Backed by the view pooling branch as a shortcut connection, (b) the Fuseption branch takes charge of learning the residual mapping. (c) The parameters of the convolutional layers are listed

pooling as a shortcut connection and adds a sibling branch termed *Fuseption* in charge of learning the underlying residual mapping.

Fuseption. As shown in Figure 1(b), the Fuseption is an Inception-style [49, 50] architecture capped above multi-view feature maps. First, following the structure of inception modules [49, 50], three lightweight cross-spatial filters with different kernel sizes, 3×3, 1×3 and 3×1, are adopted to extract different types of features. Second, the 1×1 convolution Conv6, employed above concatenated feature maps, is responsible for the merging of correlation statistics across channels and the dimension reduction as suggested by [49, 51].

Fuseption-ResNet (FRN). Inspired by the effectiveness of skip connections in ResNet [48], we take view pooling as a shortcut in addition to Fuseption as shown in Figure 1(a). As opposed to the view pooling branch which is in charge of extracting the strongest responses across views [31], the Fuseption branch is responsible for learning the underlying residual mapping. Both engaged branches reinforce each other in term of accuracy and convergence rate. On the one hand, the residual branch, Fuseption, guarantees no worse accuracy compared to just using view pooling. This is because if view pooling is optimal, the residual can be easily pulled to zeros during training. On the other hand, the shortcut branch, view pooling, greatly accelerates the convergence of learning MVDesc as illustrated in Figure 2(a). Intuitively, since the view pooling branch has extracted the essential strongest responses across views, it is easier for the Fuseption branch to just learn the residual mapping.

3.2 Learning MVDesc

Network. The network for learning MVDesc is built by putting the proposed FRN above multiple parallel feature networks. We use the feature network from MatchNet [45] as the basis, in which the bottleneck layer and the metric network are removed. The feature networks of multiple views share the same parameters of corresponding convolutional layers. The channel number of Conv6 is set to

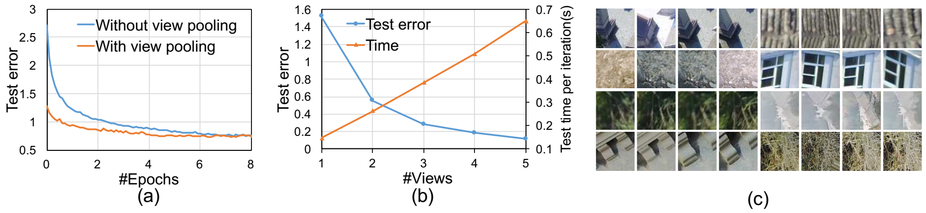


Fig. 2: (a) Test error of our MVDesc network with or without the view pooling branch. Using view pooling as a shortcut connection contributes to much faster convergence of learning. (b) Test error and forward time per iteration of MVDesc network with respect to the number of fused views. Three views are chosen as a good trade-off between accuracy and efficiency. (c) The sample multi-view patches produced by the collected SfM database for training

be the same as that of feature maps output by a feature network. The ReLU activation [52] follows each convolutional layer except the last Conv7. A layer of L2 normalization is appended after Conv7 whose channel number can be set flexibly to adjust the dimension of descriptors. The parameters of the full network are detailed in the supplemental material.

Loss. The two-tower Siamese architecture [17, 53] is adopted here for training. The formulation of the double-margin contrastive loss is used [54], *i.e.*,

$$L(\mathbf{x}_a, \mathbf{x}_b) = y \max(||\mathbf{d}_a - \mathbf{d}_b||_2 - \tau_1, 0) + (1 - y) \max(\tau_2 - ||\mathbf{d}_a - \mathbf{d}_b||_2, 0), \quad (1)$$

where $y = 1$ for positive pairs and 0 otherwise. $\mathbf{d}_a$ and $\mathbf{d}_b$ are L2-normalized MVDesc descriptors of the two sets of multi-view patches $\mathbf{x}_a$ and $\mathbf{x}_b$, output by the two towers. We set the margins $\tau_1 = 0.3, \tau_2 = 0.6$ in experiments.

View number. Unlike [31–33] using 12 or 20 views for objects, we adopt only 3 views in our MVDesc network for the description of local keypoints, which is a good tradeoff between accuracy and efficiency as shown in Figure 2(b).

Data preparation. Current available patch datasets generally lack sufficient multi-view patches for training. For example, one of the largest training sets Brown [55, 56] only possesses less than 25k 3D points with at least 6 views. Therefore, we prepare the training data similar to [20] based on the self-collected Structure-from-Motion (SfM) database. The database consists of 31 outdoor scenes of urban and rural areas captured by UAV and well reconstructed by a standard 3D reconstruction pipeline [1, 2, 57–61]. Each scene contains averagely about 900 images and 250k tracks with at least 6 projections. The multi-view patches of size 64×64 are cropped from the projections of each track according to SIFT scales and orientations [20], as displayed in Figure 2(c). A positive training pair is formed by two independent sets of triple-view patches from the same track, while a negative pair from different tracks. A total number of 10 million pairs with equal ratio of positives and negatives are evenly sampled from all the 31 scenes. We turn the patches into grayscale, subtract the intensities by 128 and divide them by 160 [45].

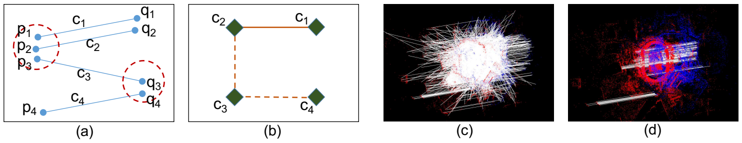


Fig. 3: (a) $c_{1\sim4}$ are four pairs of point matches. (b) The graph is defined to model the neighboring relationship between $c_{1\sim4}$. The solid/dashed lines link between compatible/incompatible neighbors. (c) 4,721 pairs of putative point matches. (d) 109 true match pairs refined by our RMBP

Training. We train the network from scratch using SGD with a momentum of 0.9, a weight decay of 0.0005 and a batch size of 256. The learning rate drops by 30% after every epoch with a base of 0.0001 using exponential decay. The training is generally done within 10 epochs.

4 Robust Matching Using Belief Propagation (RMBP)

In this section, we are devoted to enhancing the accuracy and robustness of 3D point matching. Firstly, a graphical model is defined to describe the spatial organizations of point matches. Secondly, each match pair is verified by the inference from the graphical model via belief propagation. Notably, the proposed method is complementary to the existing matching algorithms [13–15, 62].

4.1 Matching Model

It can be readily observed that inlier point correspondences generally hold *spatial proximity*. We illustrate it in Figure 3(a) where $c_1 = (\mathbf{p}_1, \mathbf{q}_1)$, $c_2 = (\mathbf{p}_2, \mathbf{q}_2)$ and $c_4 = (\mathbf{p}_4, \mathbf{q}_4)$ are three pairs of inlier correspondences. For any two pairs of inlier matches, their points in each point cloud are either spatially close to each other like $\langle \mathbf{p}_1, \mathbf{p}_2 \rangle$ and $\langle \mathbf{q}_1, \mathbf{q}_2 \rangle$ or far away from each other like $\langle \mathbf{p}_2, \mathbf{p}_4 \rangle$ and $\langle \mathbf{q}_2, \mathbf{q}_4 \rangle$ at the same time. On the contrary, outlier correspondences tend to show spatial disorders. This observation implies the probabilistic dependence between neighboring point correspondences which can be modeled by a probabilistic graph.

Formally, we first define the neighborhood of point correspondences as follows. Two pairs of point correspondences $c_i = (\mathbf{p}_i, \mathbf{q}_i)$ and $c_j = (\mathbf{p}_j, \mathbf{q}_j)$ are considered as neighbors if either $\mathbf{p}_i$ and $\mathbf{p}_j$, or $\mathbf{q}_i$ and $\mathbf{q}_j$, are mutually k-nearest neighbors, *i.e.*,

$$\max(\mathrm{rank}(\mathbf{p}_i, \mathbf{p}_j), \mathrm{rank}(\mathbf{p}_j, \mathbf{p}_i)) < k, \tag{2}$$

$$\text{or} \quad \max(\mathrm{rank}(\mathbf{q}_i, \mathbf{q}_j), \mathrm{rank}(\mathbf{q}_j, \mathbf{q}_i)) < k, \tag{3}$$

where $\mathrm{rank}(\mathbf{x}, \mathbf{y})$ denotes the rank of distance of point $\mathbf{y}$ with respect to point $\mathbf{x}$. Then, the neighboring relationship between c_i and c_j can be further divided into two categories: if Condition 2 and 3 are satisfied simultaneously, c_i and c_j are called *compatible* neighbors. They are very likely to co-exist as inlier matches. But if only one of Condition 2 or 3 is satisfied by one point pair but another pair of points in the other point cloud locate far apart from each other, *e.g.*,

$$\min(\mathrm{rank}(\mathbf{p}_i, \mathbf{p}_j), \mathrm{rank}(\mathbf{p}_j, \mathbf{p}_i)) > l, \tag{4}$$

$$\text{or} \quad \min(\mathrm{rank}(\mathbf{q}_i, \mathbf{q}_j), \mathrm{rank}(\mathbf{q}_j, \mathbf{q}_i)) > l, \tag{5}$$

c_i and c_j are called *incompatible* neighbors, as it is impossible for two match pairs breaching spatial proximity to be inliers simultaneously. The threshold parameter k in Condition 2 and 3 is set to a relatively small value, while the parameter l in Condition 4 and 5 is set to be larger than k by a considerable margin. These settings are intended to ensure sufficiently strict conditions on identifying compatible or incompatible neighbors for robustness.

Based on the spatial property of point matches stated above, an underlying graphical model is built to model the pairwise interactions between neighboring match pairs, as shown in Figure 3(a) and (b). The nodes in graphical model are first defined as a set of binary variables $\mathcal{X} = \{x_i\}$ each associated with a pair of point correspondence. $x_i \in \{0, 1\}$ indicates the latent state of being an outlier or inlier, respectively. Then the undirected edges between nodes are formed based on the compatible and incompatible neighboring relationship defined above. With the purpose of rejecting outliers, the objective here is to compute the marginal of being an inlier for each point correspondence by performing inference on the defined model.

4.2 Inference by Belief Propagation

The task of computing marginals on nodes of a cyclic network is known to be NP-hard [63]. As a disciplined inference algorithm, loopy belief propagation (LBP) provides approximate yet compelling inference on arbitrary networks [64].

In the case of our graphical network with binary variables and pairwise interactions, the probabilistic distributions of all node variables are first initialized as $[0.5, 0.5]^T$ with no prior imposed. Then the iterative message update step of a standard LBP algorithm at iteration t can be written as

$$\mathbf{m}_{ij}^{t+1} = \frac{1}{Z}\mathbf{F}_{ij}\mathbf{m}_i \prod_{k \in \partial i \setminus j} \mathbf{m}_{ki}^t. \tag{6}$$

Here, ∂i denotes the set of neighbors of node x_i and Z is the L1 norm of the incoming message for normalization. The message $\mathbf{m}_{ij}$ passed from node x_i to x_j is a two-dimensional vector, which represents the belief of x_j's probability distribution inferred by x_i. So is the constant message $\mathbf{m}_i$ passed from the observation of node x_i, which indicates the likelihood distribution of x_i given its observation measurements like descriptor similarity. The first and second components of the messages are the probabilities of being an outlier and an inlier, respectively. The product of messages is component-wise. The 2×2 matrix $\mathbf{F}_{ij}$ is the compatibility matrix of node x_i and x_j. Based on the neighboring relationship analyzed above, the compatibility matrix is supposed to favor the possibility that both nodes are inliers if they are compatible neighbors and the reverse if they are incompatible neighbors. In order to explicitly specify the bias, the compatibility matrices take two forms in the two different cases, respectively:

$$\mathbf{F}^{+} = \begin{bmatrix} 1 \; 1 \\ 1 \; \lambda \end{bmatrix} \quad \text{or} \quad \mathbf{F}^{-} = \begin{bmatrix} \lambda \; \lambda \\ \lambda \; 1 \end{bmatrix}. \tag{7}$$

The parameter λ takes a biased value greater than 1. To guarantee the convergence of LBP, Simon's condition [65] is enforced and the value of λ is thus constrained by

$$\max_{x_i \in \mathcal{X}} |\partial i| \cdot \log \lambda < 2, \tag{8}$$

in which way λ is set adaptively according to the boundary condition. The proof of the convergence's condition is detailed in the supplemental material. After convergence, the marginal distribution of node x_i is derived by

$$\mathbf{b}_i = \frac{1}{Z} \mathbf{m}_i \prod_{k \in \partial i} \mathbf{m}_{ki}, \tag{9}$$

which unifies implication from individual observations and beliefs from structured local neighborhood. After the inference, point matches with low marginals (*e.g.* < 0.5) are discarded as outliers. It greatly contributes to the matching accuracy as shown in Figure 3(d) where 109 true match pairs are refined from 4,721 noisy putative match pairs.

Complexity analysis. The complexity of LBP is known to be linear to the number of edges in the graph [66]. And the Condition 2 and 3 bound the degree of each node to be less than $2k$, so that the upper bound of RMBP's complexity is linear with the number of nodes.

5 Experiments

In this section, we first individually evaluate the proposed MVDesc and RMBP in Section 5.1 and 5.2 respectively. Then the two approaches are validated on the tasks of geometric registration in Section 5.3.

All the experiments, including the training and testing of neural networks, are done on a machine with a 8-core Intel i7-4770K, a 32GB memory and a NVIDIA GTX980 graphics card. In the experiments below, when we say putative matching, we mean finding the correspondences of points whose descriptors are mutually closest to each other in Euclidean space between two point sets. The matching is implemented based on OpenBLAS [67]. The traditional geometric descriptors [6–9] are produced based on PCL [68].

5.1 Evaluation of MVDesc

The target here is to compare the description ability of the proposed MVDesc against the state-of-the-art patch descriptors [13, 17, 18] and geometric descriptors [6–11].

Table 1: The mAPs of descriptors in the three tasks of HPatches benchmark [69]. Our MVDesc holds the top place in all the three tasks

	SIFT [13]	DeepDesc [17]	L2-Net [18]	View pooling [31]	MVDesc
Patch verification	0.646	0.716	0.792	0.883	**0.921**
Image matching	0.111	0.172	0.309	0.312	**0.325**
Patch retrieval	0.269	0.357	0.414	0.456	**0.530**

5.1.1 Comparisons with patch descriptors

Setup. We choose HPatches [69], one of the largest local patch benchmarks, for evaluation. It consists of 59 cases and 96,315 6-view patch sets. First, we partition each patch set into two subsets by splitting the 6-views into halves. Then, the 3-view patches are taken as input to generate descriptors. We set up the three benchmark tasks in [69] by reorganizing the 3-view patches and use the mean average precision (mAP) as measurement. For the patch verification task, we collect all the 96,315 positive pairs of 3-view patches and 100,000 random negatives. For the image matching task, we apply putative matching across the two half sets of each case after mixing 6,000 unrelated distractors into every half. For the patch retrieval task, we use the full 96,315 6-view patch sets each of which corresponds to a 3-view patch set as a query and the other 3-view set in the database. Besides, we mix 100,000 3-view patch sets from an independent image set into the database for distraction.

We make comparisons with the baseline SIFT [13] and the state-of-the-art DeepDesc [17] and L2-Net [18], for which we randomly choose a single view from the 3-view patches. To verify the advantage of our FRN over the widely-used view pooling [31–33] in multi-view fusion, we remove the Fuseption branch from our MVDesc network and train with the same data and configuration. All the descriptors have the dimensionality of 128.

Results. The statistics in Table 1 show that our MVDesc achieves the highest mAPs in all the three tasks. First, it demonstrates the advantage of our FRN over view pooling [31–33, 70, 40] in terms of multi-view fusion. Second, the improvement of MVDesc over DeepDesc [17] and L2-Net [18] suggests the benefits of leveraging more image views than a single one. Additionally, we illustrate in Figure 4(a) the trade-off between the mAP of the image matching task and the dimension of our MVDesc. The mAP rises but gradually saturates with the increase of dimension.

5.1.2 Comparisons with geometric descriptors

Setup. Here, we perform evaluations on matching tasks of the RGB-D dataset TUM [28]. Following [11], we collect up to 3,000 pairs of overlapping point cloud fragments from 10 scenes of TUM. Each fragment is recovered from independent RGB-D sequences of 50 frames. We detect keypoints from the fragments by SIFT3D [71] and then generate geometric descriptors, including PFH [6], FPFH [7], SHOT [8], USC [9], 3DMatch [10] and CGF [11]. Our MVDesc is derived from the projected patches of keypoints in three randomly-selected camera views. For easier comparison, two dimensions, 32 and 128, of MVDesc (MVDesc-32 and

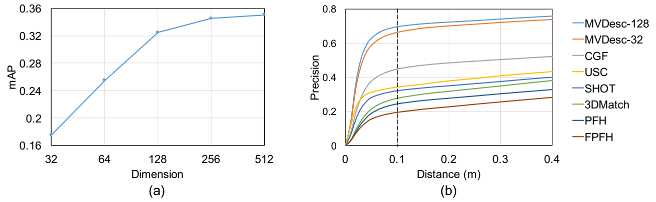


Fig. 4: (a) The trade-off of mAP versus dimension of our MVDesc on the HPatches benchmark [69]; (b) The change of matching precisions *w.r.t.* the varying threshold of point distances on the TUM dataset [28]. The 128- and 32-dimensional MVDesc rank first and second at any threshold

Table 2: Precisions and recalls of matching on the TUM dataset [28] when the threshold of points' distances equals to 0.1 meters. The average time taken to encode 1,000 descriptors is also compared. Our MVDesc hits the best in terms of precision, recall and efficiency

	CGF [11]	FPFH [7]	PFH [6]	SHOT [8]	3DMatch [10]	USC [9]	MVDesc	
Dim.	32	33	125	352	512	1980	32	128
Precision	0.447	0.194	0.244	0.322	0.278	0.342	0.664	**0.695**
Recall	0.215	0.229	0.265	0.093	0.114	0.026	0.523	**0.580**
Time (s)	7.60	1.49	14.40	0.29	2.60	0.73	**0.22**	0.23

MVDesc-128) are adopted. Putative matching is applied to all the fragment pairs to obtain correspondences. Following [11], we measure the performance of descriptors by the fraction of correspondences whose ground-truth distances lie below the given threshold.

Results. The precision of point matches *w.r.t.* the threshold of point distances is depicted in Figure 4(b). The MVDesc-128 and MVDesc-32 rank first and second in precision at any threshold, outperforming the state-of-the-art works by a considerable margin. We report in Table 2 the precisions and recalls when setting the threshold to 0.1 meters and the average time of producing 1,000 descriptors. Producing geometric descriptors in general is slower than MVDesc due to the cost of computing local histograms, although the computation has been accelerated by multi-thread parallelism.

5.2 Evaluation of RMBP

Setup. To evaluate the performance of outlier rejection, we compare RMBP with RANSAC and two state-of-the-art works - Sparse Matching Game (SMG) [42] and Locality Preserving Matching (LPM) [43]. All the parameters of methods have been tuned to give the best results. We match 100 pairs of same-scale point clouds from 20 diverse indoor and outdoor scenes of TUM [27], ScanNet [65] and EPFL [38] datasets. We keep a constant number of inlier correspondences and continuously add outlier correspondences for distraction. The evaluation uses the metrics: the mean precisions and recalls of outlier rejection and

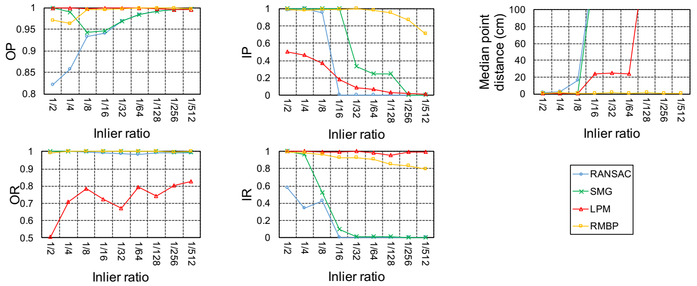


Fig. 5: The mean precisions and recalls of outlier rejection (OP, OR) and inlier selcection (IP, IR), as well as median point distance after registration, *w.r.t.* the inlier ratio. RMBP performs well in all metrics and at all inlier ratios while RANSAC, SMG [42] and LPM [43] fail to give valid registrations when the inlier ratio drops below $\frac{1}{8}$, $\frac{1}{8}$ and $\frac{1}{64}$, respectively

inlier selection. Formally, we write $OP = \frac{\#true\ rejections}{\#rejections}$, $OR = \frac{\#true\ rejections}{\#outliers}$, $IP = \frac{\#kept\ inliers}{\#kept\ matches}$, $IR = \frac{\#kept\ inliers}{\#inliers}$. We also estimate the transformations from kept matches using RANSAC and collect the median point distance after registration.

Results. The measurements with respect to the inlier ratio are shown in Fig. 5. First, RMBP is the only method achieving high performance in all metrics and at all inlier ratios. Second, RANSAC, SMG and LPM fail to give valid registrations when the inlier ratio drops below $\frac{1}{8}$, $\frac{1}{8}$ and $\frac{1}{64}$, respectively. They obtain high OP and OR but low IP or IR when the inlier ratio is smaller than $\frac{1}{16}$, because they tend to reject almost all the matches.

5.3 Geometric Registration

In this section, we verify the practical usage of the proposed MVDesc and RMBP by the tasks of geometric registration. We operate on point cloud data obtained from two different sources: the point clouds scanned by RGB-D sensors and those reconstructed by multi-view stereo (MVS) algorithms [72].

5.3.1 Registration of scanning data

Setup. We use the task of loop closure as in [10, 11] based on the dataset ScanNet [73], where we check whether two overlapping sub-maps of an indoor scene can be effectively detected and registered. Similar to [10, 11], we build up independent fragments of 50 sequential RGB-D frames from 6 different indoor scenes of ScanNet [73]. For each scene, we collect more than 500 fragment pairs with labeled overlap obtained from the ground truth for registration.

The commonly-used registration algorithm, putative matching plus RANSAC, is adopted in combination with various descriptors [6–11]. The proposed RMBP serves as an optional step before RANSAC. We use the same metric as [10, 11], *i.e.*, the precision and recall of registration of fragment pairs. Following [10], a

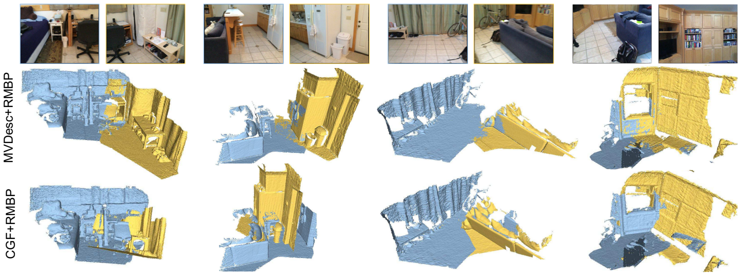


Fig. 6: Challenging cases of loop closures from the ScanNet dataset [73]. The images in the top row indicate very limited overlap shared by the fragment pairs. Our MVDesc-32+RMBP succeeds in the registration of these cases while the top-performing geometric descriptor CGF-32 [11] fails no matter whether RMBP is employed

Table 3: The quantitative comparisons of 3D descriptors on registration of the ScanNet dataset [73]. The superscript * means the proposed RMBP is applied. The RMBP generally lifts the precisions and recalls of registration for almost all the descriptors. Our MVDesc well surpasses the state-of-the-art works in recall with comparable precision and run-time of registration

	CGF [11]	FPFH [7]	PFH [6]	SHOT [8]	3DMatch [10]	USC [9]	MVDesc	
Dim.	32	33	125	352	512	1980	32	128
Precision	0.914	0.825	0.866	0.875	0.890	0.790	0.865	0.910
Precision*	0.927	0.856	0.864	0.928	0.934	0.795	0.892	0.906
Recall	0.350	0.119	0.147	0.178	0.185	0.124	0.421	0.490
Recall*	0.419	0.272	0.338	0.198	0.145	0.157	0.482	0.513
Time (s)	0.5	0.5	0.7	1.8	2.4	8.4	0.5	0.7

registration is viewed as true positive if the estimated Euclidean transformation yields more than 30% overlap between registered fragments and transformation error less then $0.2m$. We see a registration as positive if there exist more than 20 pairs of point correspondences after RANSAC.

Results. The precisions, recalls and the average time of registration per pair are reported in Table 3. Our MVDesc-32 and MVDesc-128 both surpass the counterparts by a significant margin in recall while with comparable precision and efficiency. Our versatile RMBP well improves the precisions for 6 out of 8 descriptors and lifts the recalls for 7 out of 8 descriptors. The sample registration results of overlap-deficient fragments are visualized in Figure 6.

Indoor reconstruction. The practical usage of MVDesc is additionally evaluated by indoor reconstruction of the ScanNet dataset [73]. We first build up reliable local fragments through RGB-D odometry following [74, 75] and then globally register the fragments based on [15]. The RMBP is applied for outlier filtering. The FPFH [7] used in [15] is replaced by SIFT [13], CGF-32 [11] or MVDesc-32 to establish correspondences. We also test the collaboration of CGF-32 and MVDesc-32 by combining their correspondences. Our MVDesc-32 contributes to visually compelling reconstruction as shown in Figure 7(a). And

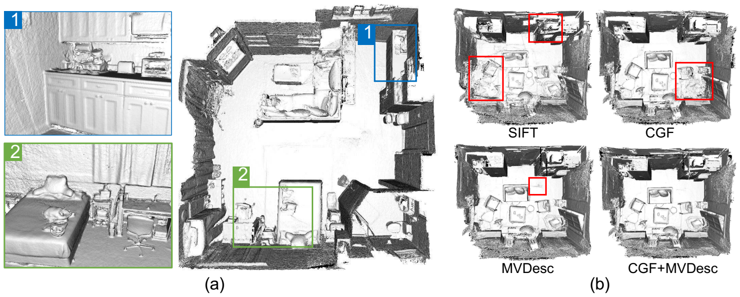


Fig. 7: (a) A complete reconstruction of an apartment from ScanNet [73] using our MVDesc-32. (b) The reconstructions using SIFT [13], CGF-32 [11] and MVDesc-32. The collaboration of MVDesc-32 and CGF-32 yields the best reconstruction as shown in the last cell of (b)

we find that MVDesc-32 functions as a solid complement to CGF-32 as shown in Figure 7(b), especially for the scenarios with rich textures.

5.3.2 Registration of Multi-View Stereo (MVS) data

Setup. Aside from the scanning data, we run registration on the four scenes of the MVS benchmark EPFL [39]. First, we split the cameras of each scene into two clusters in space highlighting the difference between camera views, as shown in Figure 8. Then, the ground-truth camera poses of each cluster are utilized to independently reconstruct the dense point clouds by the MVS algorithm [72]. Next, we detect keypoints by SIFT3D [71] and generate descriptors [6–11] for each point cloud. The triple-view patches required by MVDesc-32 are obtained by projecting keypoints into 3 visible image views randomly with occlusion test by ray racing [76]. After, the correspondences between the two point clouds of each scene are obtained by putative matching and then RMBP filtering. Finally, we register the two point clouds of each scene based on FGR [15] using estimated correspondences.

Results. Our MVDesc-32 and RMBP help to achieve valid registrations for all the four scenes, whilst none of the geometric descriptors including CGF [11], 3DMatch [10], FPFH [7], PFH [6], SHOT [8] and USC [9] do, as shown in Figure 8. It is found that the failure is mainly caused by the geometrically symmetric patterns of the four scenes. We show the correspondences between CGF-32 features [11] in Figure 9 as an example. The putative matching has resulted in a large number of ambiguous correspondences between keypoints located at the symmetric positions. And in essence, our RMBP is incapable of resolving the ambiguity in such cases though, because the correspondences in a symmetric structure still adhere to the geometric consistency. Ultimately, the ambiguous matches lead to the horizontally-flipped registration results as shown in Figure 8. At least in the EPFL benchmark [39], the proposed MVDesc descriptor shows superior ability of description to the geometric ones.

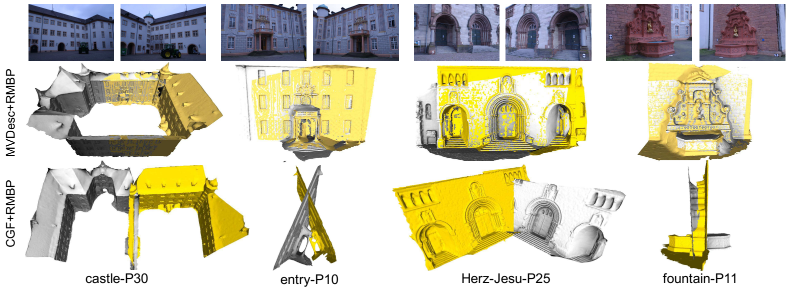


Fig. 8: Registrations of models of EPFL benchmark [39]. Given the same keypoints, our MVDesc-32+RMBP accomplishes correct registrations while CGF-32 [11]+RMBP fails due to the symmetric ambiguity of the geometry

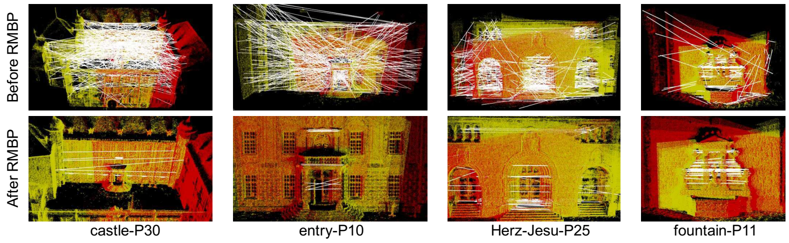


Fig. 9: Spurious correspondences of CGF features [11] before and after RMBP filtering. The two point clouds of [39] (colored in yellow and red) have been overlaid together by the ground truth transformation for visualization. Although our RMBP has eliminated most of the unorganized false matches, it is incapable of rejecting those ambiguous outliers arising from the ambiguity of the symmetric geometry

6 Conclusion

In this paper, we address the correspondence problem for the registration of point clouds. First, a multi-view descriptor, named MVDesc, has been proposed for the encoding of 3D keypoints, which strengthens the representation by applying the fusion of image views [31–33] to patch descriptor learning [17–21]. Second, a robust matching method, abbreviated as RMBP, has been developed to resolve the rejection of outlier matches by means of efficient inference through belief propagation [64] on the defined graphical matching model. Both approaches have been validated to be conductive to forming point correspondences of better quality for registration, as demonstrated by the intensive comparative evaluations and registration experiments [6–11, 15, 17, 18, 62].

Acknowledgement. This work is supported by T22-603/15N, Hong Kong ITC PSKL12EG02 and The Special Project of International Scientific and Technological Cooperation in Guangzhou Development District (No. 2017GH24).

References

1. Zhu, S., Zhang, R., Zhou, L., Shen, T., Fang, T., Tan, P., Quan, L.: Very large-scale global sfm by distributed motion averaging. In: CVPR. (2018)
2. Zhang, R., Zhu, S., Shen, T., Zhou, L., Luo, Z., Fang, T., Quan, L.: Distributed very large scale bundle adjustment by global camera consensus. PAMI (2018)
3. Dissanayake, M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (slam) problem. IEEE Transactions on Robotics and Automation **17**(3) (2001) 229–241
4. Fraundorfer, F., Scaramuzza, D.: Visual odometry: Part ii: Matching, robustness, optimization, and applications. IEEE Robotics & Automation Magazine **19**(2) (2012) 78–90
5. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. PAMI **21**(5) (1999)
6. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: IROS. (2008)
7. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: ICRA. (2009)
8. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: ECCV. (2010)
9. Tombari, F., Salti, S., Di Stefano, L.: Unique shape context for 3d data description. In: Proceedings of the ACM workshop on 3D object retrieval. (2010)
10. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: CVPR. (2017)
11. Khoury, M., Zhou, Q.Y., Koltun, V.: Learning compact geometric features. In: ICCV. (2017)
12. Deng, H., Birdal, T., Ilic, S.: Ppfnet: Global context aware local features for robust 3d point matching. arXiv preprint (2018)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2) (2004) 91–110
14. Gold, S., Lu, C.P., Rangarajan, A., Pappu, S., Mjolsness, E.: New algorithms for 2d and 3d point matching: Pose estimation and correspondence. In: Advances in Neural Information Processing Systems. (1995)
15. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: ECCV. (2016)
16. Georgakis, G., Karanam, S., Wu, Z., Ernst, J., Kosecka, J.: End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. arXiv preprint (2018)
17. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: ICCV. (2015)
18. Tian, B.F.Y., Wu, F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: CVPR. (2017)
19. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: CVPR. (2015)
20. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: ECCV. (2016)
21. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: BMVC. (2016)
22. Wu, C., Clipp, B., Li, X., Frahm, J.M., Pollefeys, M.: 3d model matching with viewpoint-invariant patches (vip). In: CVPR. (2008)

23. Chu, J., Nie, C.m.: Multi-view point clouds registration and stitching based on sift feature. In: ICCRD. (2011)
24. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. TO **36**(3) (2017) 24
25. Endres, F., Hess, J., Sturm, J., Cremers, D., Burgard, W.: 3-d mapping with an rgb-d camera. IEEE Transactions on Robotics **30**(1) (2014) 177–187
26. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: 3DV. (2017)
27. Handa, A., Whelan, T., McDonald, J., Davison, A.: A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: ICRA. (2014)
28. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: IROS. (2012)
29. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. In: SIGGRAPH. (2006)
30. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: ICCV. (2009)
31. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: ICCV. (2015)
32. Wang, C., Pelillo, M., Siddiqi, K.: Dominant set clustering and pooling for multi-view 3d object recognition. In: BMVC. (2017)
33. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: CVPR. (2016)
34. Besl, P.J., McKay, N.D., et al.: A method for registration of 3-d shapes. PAMI **14**(2) (1992) 239–256
35. Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: Comparing icp variants on real-world data sets. Autonomous Robots **34**(3) (2013)
36. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: 3DIM. (2001)
37. Yang, J., Li, H., Campbell, D., Jia, Y.: Go-icp: a globally optimal solution to 3d icp point-set registration. PAMI **38**(11) (2016) 2241–2254
38. Briales, J., Gonzalez-Jimenez, J.: Convex global 3d registration with lagrangian duality. In: CVPR. (2017)
39. Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR. (2008)
40. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. arXiv preprint (2016)
41. Albarelli, A., Bulo, S.R., Torsello, A., Pelillo, M.: Matching as a non-cooperative game. In: ICCV. (2009)
42. Rodolà, E., Albarelli, A., Bergamasco, F., Torsello, A.: A scale independent selection process for 3d object recognition in cluttered scenes. IJCV **102**(1-3) (2013)
43. Jiayi, M., Ji, Z., Hanqi, G., Junjun, J., Huabing, Z., Yuan, G.: Locality preserving matching. In: IJCAI. (2017)
44. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. IJCV **19**(1) (1996) 57–91
45. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: CVPR. (2015)

46. Anastasiya, M., Dmytro, M., Filip, R., Jiri, M.: Working hard to know your neighbor's margins: Local descriptor learning loss. In: NIPS. (2017)
47. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV. (2016)
48. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
49. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2015)
50. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. (2016)
51. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint (2013)
52. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint (2015)
53. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR. (2005)
54. Lin, J., Morère, O., Veillard, A., Duan, L.Y., Goh, H., Chandrasekhar, V.: Deephash for image instance retrieval: Getting regularization, depth and fine-tuning right. In: ICMR. (2017)
55. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. IJCV **80**(2) (2008) 189–210
56. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: ICCV. (2007)
57. Zhou, L., Zhu, S., Shen, T., Wang, J., Fang, T., Quan, L.: Progressive large scale-invariant image matching in scale space. In: ICCV. (2017)
58. Shen, T., Zhu, S., Fang, T., Zhang, R., Quan, L.: Graph-based consistent matching for structure-from-motion. In: ECCV. (2016)
59. Zhu, S., Shen, T., Zhou, L., Zhang, R., Wang, J., Fang, T., Quan, L.: Parallel structure from motion from local increment to global averaging. arXiv preprint arXiv:1702.08601 (2017)
60. Zhang, R., Zhu, S., Fang, T., Quan, L.: Distributed very large scale bundle adjustment by global camera consensus. In: ICCV. (2017)
61. Li, S., Siu, S.Y., Fang, T., Quan, L.: Efficient multi-view surface refinement with adaptive resolution control. In: ECCV. (2016)
62. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In: ECCV. (2008)
63. Cooper, G.F.: The computational complexity of probabilistic inference using bayesian belief networks. Artificial intelligence **42**(2-3) (1990) 393–405
64. Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: UAI. (1999)
65. Tatikonda, S.C., Jordan, M.I.: Loopy belief propagation and gibbs measures. In: UAI. (2002)
66. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. Exploring artificial intelligence in the new millennium (2003) 239–269
67. Zhang, X., Wang, Q., Werner, S., Zaheer, C., Chen, S., Luo, W.: http://www.openblas.net/
68. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: ICRA. (2011)
69. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: CVPR. (2017)

70. Huang, H., Kalogerakis, E., Chaudhuri, S., Ceylan, D., Kim, V.G., Yumer, E.: Learning local shape descriptors from part correspondences with multiview convolutional networks. TOG **37**(1) (2018) 6
71. Flint, A., Dick, A., Van Den Hengel, A.: Thrift: Local 3d structure recognition. In: Digital Image Computing Techniques and Applications. (2007)
72. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV. (2016)
73. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR. (2017)
74. Kerl, C., Sturm, J., Cremers, D.: Robust odometry estimation for rgb-d cameras. In: ICRA. (2013)
75. Choi, S., Zhou, Q.Y., Koltun, V.: Robust reconstruction of indoor scenes. In: CVPR. (2015)
76. Wald, I., Slusallek, P., Benthin, C., Wagner, M.: Interactive rendering with coherent ray tracing. In: Computer graphics forum. (2001)