SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2023, Vol. 66 172101:1–172101:17
https://doi.org/10.1007/s11432-022-3604-6

Hybrid3D: learning 3D hybrid features with point clouds and multi-view images for point cloud registration

Bangbang YANG[1], Zhaoyang HUANG[2], Yijin LI[1], Han ZHOU[1], Hongsheng LI[2], Guofeng ZHANG[1] & Hujun BAO[1*]

[1]*State Key Lab of CAD&CG, Zhejiang University, Hangzhou* 310058*, China;*
[2]*The Chinese University of Hong Kong, Hong Kong* 999077*, China*

Received 18 March 2022/Revised 6 June 2022/Accepted 7 September 2022/Published online 29 June 2023

Abstract In recent years, point cloud registration has achieved great success by learning geometric features with deep learning techniques. However, existing approaches that rely on pure geometric context still suffer from sensor noise and geometric ambiguities (e.g., flat or symmetric structure), which limit their robustness to real-world scenes. When 3D point clouds are constructed by RGB-D cameras, we can enhance the learned features with complementary texture information from RGB images. To this end, we propose to learn a 3D hybrid feature that fully exploits the multi-view colored images and point clouds from indoor RGB-D scene scans. Specifically, to address the discrepancy of 2D-3D observations, we design to extract informative 2D features from image planes and take only these features for fusion. Then, we utilize a novel soft-fusion module to associate and fuse hybrid features in a unified space while alleviating the ambiguities of 2D-3D feature binding. Finally, we develop a self-supervised feature scoring module customized for our multi-modal hybrid features, which significantly improves the keypoint selection quality in noisy indoor scene scans. Our method shows competitive registration performance with previous methods on two real-world datasets.

Keywords point cloud registration, cross-modal feature fusion, multi-view feature fusion, computer vision, deep learning

Citation Yang B B, Huang Z Y, Li Y J, et al. Hybrid3D: learning 3D hybrid features with point clouds and multi-view images for point cloud registration. Sci China Inf Sci, 2023, 66(7): 172101, https://doi.org/10.1007/s11432-022-3604-6

1 Introduction

Point cloud registration, which aims to align two 3D point cloud fragments into a unified coordinate, is widely utilized in many 3D vision applications, such as 3D reconstruction [1, 2], pose estimation [3], loop closing [4], and mapping [5]. The key component of this task is to establish reliable correspondences between partial scans, which practically determines the registration quality when solving rigid transformation [6].

Over the decades, enormous efforts have been made to construct point correspondences by matching with geometry features [7–13]. However, in practice, scene-scanned point cloud is often noisy and incomplete, and pure geometry cues face the ambiguity issue (e.g., flat or symmetric geometry), which inevitably introduces mismatches that degrade the registration performance. Fortunately, in scene-level applications such as robotics and 3D scene reconstruction, depth sensors and LiDAR scanners are often accompanied by RGB cameras, and many laser scanners (e.g., Faro Focus 3D-X330 used by Redwood [14]) also support simultaneous overlay photo capture. In cases where the scene is captured by RGB-D cameras, we can simultaneously obtain multi-view images and depth. This rich texture information in images can improve the discrimination of descriptors if we can coalesce the point cloud and the images.

In this paper, we take a further step toward complementing 3D geometric features with multi-view images. By extracting features in 2D image planes and 3D space, we successfully find out discriminative

* Corresponding author (email: bao@cad.zju.edu.cn)

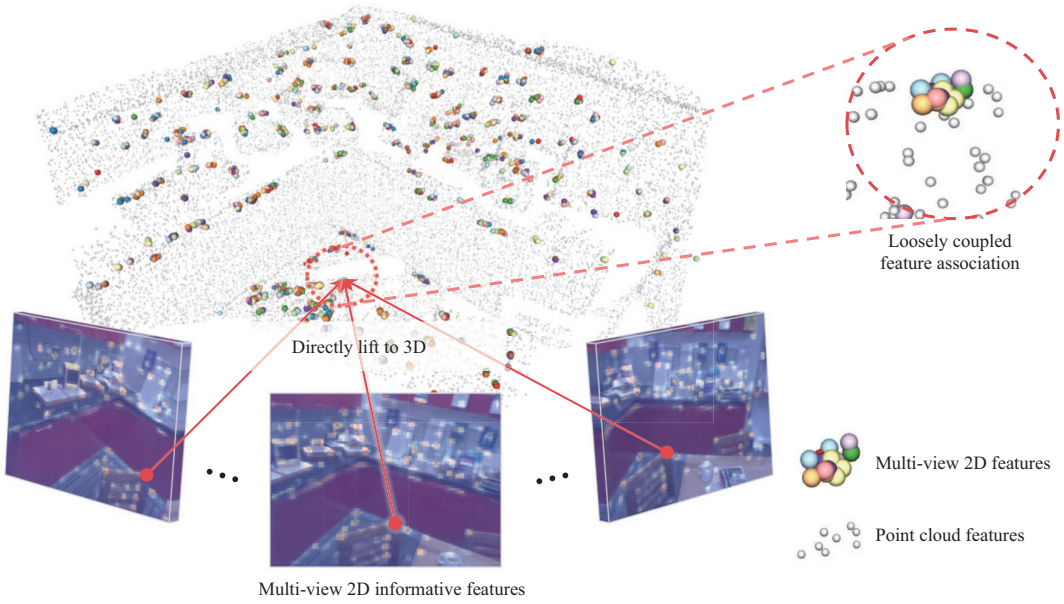

Figure 1 (Color online) Loosely coupled feature association. Our Hybrid3D lifts 2D features to 3D space based on pixel depth and camera parameters. In this way, multi-view 2D features and 3D point cloud features are loosely coupled in a unified 3D space.

patterns that only exist in the texture domain (e.g., textured poster with a flat geometry) or geometry domain (e.g., furniture or corners with a poor texture). As illustrated in Figures 1 and 2, by fusing multi-view and multi-modal features, we obtain a strong hybrid descriptor and detector, which takes advantage of both 2D and 3D feature learning, and achieves superior registration performance. However, effectively fusing multi-view image features and 3D geometric features while improving registration performance is non-trivial, as we address below.

Discrepancy of 2D-3D observations. In general, visual observations in the image domain are much denser than geometry observations in the point cloud domain; i.e., an image of 640×480 resolution yields almost 30000 pixel-wise visual features, which can be tens of large when considering multi-view images for one point cloud fragment, and soon become unaffordable if directly binding them to geometric features in 3D space. To solve this problem, we propose to select only a few informative features from the multi-view image domain and then project them to 3D space. Intuitively, the selected 2D features should be consistent among different views and discriminative to different 3D points. Previous 2D image feature learning frameworks [15–18] mainly rely on pairwise 2D correspondences for supervision, which shows inferior consistency in multi-view images. By contrast, we train our network with a novel clustering and projection strategy, which fully leverages multi-view observations by grouping and filtering all the 2D proposals in a unified 3D coordinate, and repeatedly yields pseudo-labels for 2D keypoint supervision. In this way, pixels belonging to the same 3D point in multi-view images are encouraged to produce coherent predictions. Thus, the learned features are more robust and can benefit 3D registration.

Ambiguity of 2D-3D feature association. To perform multi-modal feature fusion, we need to determine the relationship/correspondence of 2D-3D features in advance. However, many technical challenges hinder feature fusion in multi-view cases. First, a 3D point can be observed in an indefinite number of images, which is regarded as a one-to-many mapping problem and stumps the methods that directly concatenate 2D-3D features [19–23]. Second, hard 2D-3D correspondence assignments can generate misleading hybrid features once the correspondence is erroneous due to sensor noises and geometric errors (e.g., pose or calibration error). Third, due to the different spatial densities of 2D and 3D features (i.e., dense 3D features and sparse 2D features), a naïvely feature association might lead to degraded performance (see Subsections 3.3 and 4.4). To tackle all these challenges, we propose a novel loosely-coupled feature association, as illustrated in Figure 1. Instead of manually finding relationships with geometry projection or pixel-wise assignment, we first generate a hybrid point cloud by mixing up 2D features and 3D point features in a unified space, so the cross-modality features are loosely-coupled based on their coordinates. Then, we extract 3D hybrid features with soft feature fusion, where the balanced fusion strategy allows 3D points to fully interact with surrounding 2D features while alleviating the problem

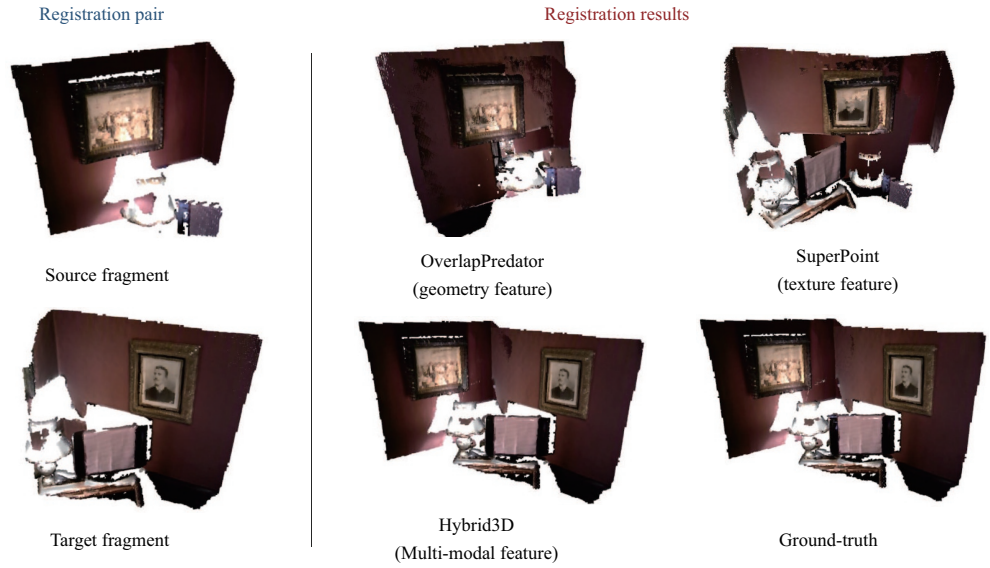

Figure 2 (Color online) Geometric registration comparison with different features. We show a challenging example of a RANSAC-based geometric registration from the Redwood dataset. The SOTA geometric feature-based method, i.e., OverlapPredator [24] and the texture feature-based method SuperPoint [17] both struggle to register the two fragments due to the geometric ambiguity (flat walls) and insufficient texture. By contrast, our Hybrid3D leverages multi-modal information and faithfully registers the fragments.

caused by unbalanced feature densities (see Subsection 4.4).

Absence of multi-modal keypoint selection. The mainstream point cloud registration for indoor scene scans usually obtains matches with random keypoints, so they require a large number of keypoints (e.g., 5000 keypoints in most studies) to ensure the registration quality, which is inefficient for practical applications. Moreover, the sampled keypoints at non-saliency regions (e.g., flat planes) would degrade the performance [11]. However, existing approaches applicable for noisy point clouds either rely on pairwise inputs [24, 25], or only focus on the 3D geometry saliency which is still far from satisfactory when only a small number of keypoints is allowed [11, 26]. None of the existing keypoint detectors are suitable for such multi-view and multi-modal feature learning diagrams. To this end, we customize a lightweight 3D scoring module for our multi-modal features. Instead of finding saliency keypoints from the pure geometry context or 2D image domains, our scoring approach directly qualifies learned descriptors, and learns to prioritize hybrid descriptors self-supervised by on-the-fly descriptor discriminative statistics. Thanks to the 3D scoring module, we can achieve state-of-the-art registration performance with only a few keypoints compared with the existing methods. The source code of our method is available at https://github.com/zju3dv/hybrid3d/.

In summary, our key contributions are as follows:

• We propose a 3D hybrid feature learning framework for point clouds built by RGB-D cameras, which effectively complements point features with information from multi-view images. Our Hybrid3D achieves competitive registration performance with previous methods on two real-world datasets.

• To fill the gap of 2D-3D feature densities, we propose a 2D feature proposal (2DFP) module to select informative features at image planes. Guided by a self-supervised clustering and projection strategy, the 2DFP achieves cross-view robustness.

• We propose a novel 3D soft fusion (3DSF) module that fuses multi-modal features with a loosely coupled 2D-3D association, which tackles several challenges of 2D-3D feature binding.

• We propose a self-supervised keypoint scoring module that is customized for our multi-modal features and significantly improves keypoint selection quality in noisy indoor scenes.

2 Related work

2.1 3D keypoint description and matching

Early studies are mainly hand-crafted [9], e.g., using histogram statistic [27–30] or neighbor relationships [31] to produce 3D local descriptors [9], which generally fail to work when feeding with scene-

scanned point clouds [32] due to their poor generalization and robustness. In recent years, an increasing number of researchers have focused on learning 3D local descriptors and keypoint matching with deep neural networks. The most common studies [4, 11, 33–38] use point cloud patches as the network input and build upon existing point networks [39–42]. 3DMatch [32] and 3DSmoothNet [43] leverage a voxel grid representation and train a 3D-ConvNet-based descriptors for 3D feature matching, which not only outperforms the PointNet-like approaches but also sacrifices the inference speed. Choy et al. proposed FCGF [44] that adopt sparse tensor representation [45] for point clouds. Ao et al. [46], Spezialetti et al. [47] and Chen et al. [48] proposed to improve the equivariance of geometry descriptor or matching. Choy et al. [6] and Bai et al. [49] boosted the existing RANSAC-based correspondence selection with learning-based spatial context. Inspired by Sarlin et al. [50], Huang et al. [24] and Yu et al. [25] leveraged a cross-attention mechanism to improve the matching quality in low-overlapping cases. However, these lines of work rely on pairwise point clouds as the input and require duplicated network forwarding for each point cloud with multiple matching candidates, which might not be applicable in real applications as matching candidates grow. Besides, some studies also tend to learn 3D descriptors with multi-view rendered geometry patches [51, 52] or image features [53]. Recently, Banani et al. proposed to learn point cloud registration self-supervised with visual cues [54, 55], but lacked proper ways to handle multi-view observations. However, in terms of the network input and feature extraction, none of them tried to jointly fuse multi-view colored images with 3D point cloud for descriptor learning.

2.2 3D keypoint detectors

In contrast to the vigorous development of 3D keypoint description and matching, existing 3D keypoint detectors [31, 56, 57] are mainly hand-crafted, which are susceptible to sensor noise and hard to generalize on scene-scanned point clouds [26]. Recently, a few studies [11, 26, 58] have focused on detecting keypoints in a data-driven manner. USIP [26] is the first to explicitly learn keypoint detection, which outperforms traditional detectors in outdoor scenes and CAD models but degenerates in real-world indoor scanned data [11]. D3Feat [11] makes use of a spatial and channel-wise maximum of descriptors to predict the scores and is supervised by online feature matching results during training. In this paper, we also borrow such an idea from D3Feat [11]. However, existing 3D keypoint detectors only make use of pure geometry context, but they are not applicable to leverage rich contexts in 2D image domains. By contrast, our keypoint scoring module directly qualifies the distinctiveness of learned descriptors, which can be naturally customized for multi-modal features.

2.3 2D-3D feature fusion

Colored images and point clouds both contain rich information on different domains. Therefore, many studies in 3D vision tend to fuse these two kinds of observations. In the field of object detection and semantic segmentation, some studies [20, 23, 59–64] focused on fusing 2D and 3D features using various network architectures, and then predicted objects or performed segmentation upon the fused features. To achieve online scene segmentation (i.e., real-time 3D scene reconstruction along with semantic segmentation), Zhang et al. [63] and Huang et al. [64] also fused multi-modal and cross-frame features with tree-based 3D convolution or supervoxel convolution, but they neglected the undetermined issue of 2D-3D association (i.e., indefinite number of observations and misleading correlation due to sensor noise). In object pose estimation, DenseFusion [19] and PVN3D [22] use PointNet++ [40] to process 3D data, and then concatenate image features to point cloud features for dense fusion. Other studies [21, 65] leveraged sparse point clouds to enhance image-based depth completion. In 3D local feature learning, 3DTNet [66] concatenates and fuses multi-scale 2D/3D learned features in a single view, and only evaluates between frames instead of scenes. Besides, one class of related studies [67–69] contributes to matching 2D and 3D cross-domain features, which is different from ours that focus on learning cross-domain feature fusion. Nevertheless, existing studies either performed single-view 2D-3D feature fusion [19, 20, 22, 66], or used RGB images with fixed [60] or limited views [61, 62] for multi-view 2D-3D feature fusion. None of them tackles the problem of jointly fusing features across arbitrary 2D observations and point clouds in real-world scene scans.

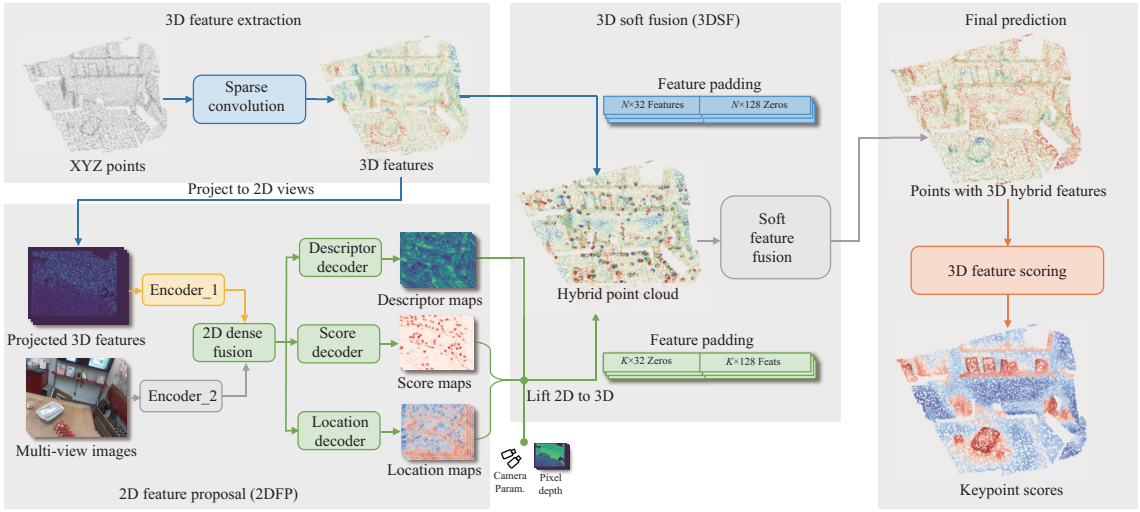

Figure 3 (Color online) Hybrid3D framework pipeline. Given a 3D point cloud and corresponding RGB images, we first use a sparse convolution to extract 3D features on the point cloud and project them to 2D image planes. Then, we abstract sparse 2D features from the projected 3D features and RGB images with the 2D feature proposal module (2DFP). After that, we lift the multi-view 2D features to a unified 3D space via pixel depth and camera parameters, and generate a hybrid point cloud by softly associating 2D features and 3D features in 3D space. Then, we exploit the 3D soft fusion module (3DSF) to extract 3D hybrid features from the hybrid point cloud. Finally, we utilize a 3D feature scoring module to predict scores for keypoint selection.

3 Methods

3.1 Overview

We propose a multi-modal 3D feature detection and description framework, named Hybrid3D. As illustrated in Figure 3, our framework consists of four stages. First, we extract 3D geometric features with sparse convolution [44] (upper-left of Figure 3). Second, we dig out representative 2D features from multi-view RGB images and projected geometric features on 2D image views within the 2DFP module (lower-left of Figure 3). Third, we lift all these 2D feature proposals to 3D space via pixel depth and camera poses, and generate a hybrid point cloud that contains 2D-3D mixed features. It is noteworthy that we do not explicitly associate 2D features to a certain 3D point or vice versa, but utilize a 3DSF module to adaptively handle all scattered multi-modal features and distill 3D hybrid features for registration (middle of Figure 3). Forth, to boost the feature matching efficiency, we leverage a lightweight 3D scoring module (right side of Figure 3), which directly regresses the distinctiveness/quality of the hybrid feature and is the key to achieving superior registration quality with only a few keypoints.

3.2 Multi-view 2D feature proposal

For indoor scene scans, a point cloud is often reconstructed by multi-view 2D observations (e.g., RGB-D frames), so it is inefficient to lift all the 2D dense image features to 3D space as former studies [19,23,66]. Moreover, 2D features lie in texture-less areas (e.g., white walls or solid color carpets) and generally cannot provide valuable information for feature matching. To this end, we settle on selecting informative sparse features from multi-view images by 2DFP module, and only lift these sparse features to 3D space. In this way, we can save computation resources while preserving sufficient texture information for multi-modal feature fusion.

3.2.1 *2DFP module architecture*

As illustrated in the lower-left part of Figure 3, the 2DFP module takes both inputs from projected 3D point features and RGB images, and forwards with two independent encoders, followed by a pixel-wise dense fusion to obtain a multi-modal feature map. Specifically, we concatenate pixel-wise feature maps from two modalities and then use a convolutional neural network (CNN) layer to fuse features (see the supplementary for network details). Then, we use three output decoders to regress descriptor maps, score maps, and location maps. Similar to previous studies [18,70], these output maps are all in a downscale (1/8) image domain for better efficiency and convergence. (1) The score map indicates the confidence of

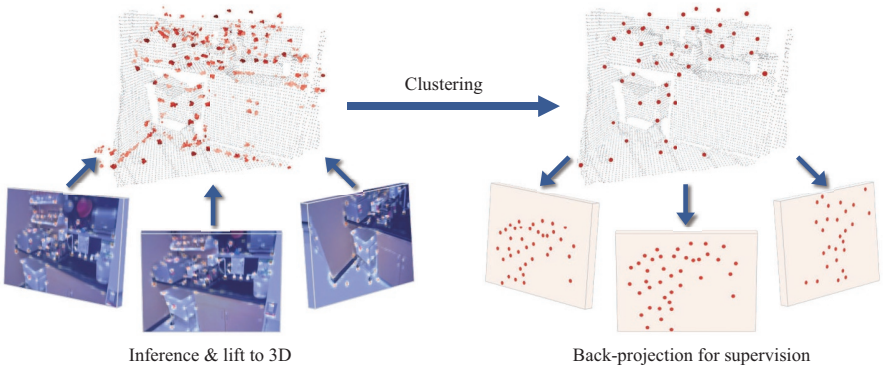

Figure 4 (Color online) Clustering and projection strategy. In 2DFP training, we lift 2D predicted keypoints from multi-view images to 3D space and cluster them to several 3D centers, then back-project these centers for 2D supervision.

keypoints. (2) The location map predicts the relative keypoint offset to each score map's pixel center. By incorporating score maps (with non-maximum-suppression (NMS) keypoint selection) and location maps (with local offset added to the NMS's resulting coordinate) [18, 70], we can obtain sub-pixel level keypoints. Besides, we apply a threshold clipping and a non-maximum-suppression to select salience keypoints on score maps, and utilize the location map to get the accurate 2D keypoint coordinates. (3) The descriptor map contains the fused multi-modal features, where the interpolated descriptors on the 2D keypoints will be lifted to 3D space according to the corresponding pixel depth and camera parameters.

3.2.2 *2DFP supervision*

We now seek a proper way to take advantage of multi-view contexts for supervision. Most of the existing studies [15–18] employed contrastive learning with pairwise correspondence. However, pairwise supervision may lead to feature drifting along with the correspondence transitivity [71]. In contrast, we leverage the geometric relationships among multi-view observations to set up a clustering and projection strategy in a self-supervised fashion. By encouraging corresponding pixels in multi-view images to generate consistent features, our 2DFP can infer more robust 2D features with view-point variation, which benefits point cloud registration.

Cross-view supervision. We use overlapped point cloud fragment pairs with corresponding images as training input. As illustrated in Figure 4, for a specific fragment pair, we first uniformly sample $\mathcal{M}$ image views from the corresponding image sequence. Then we use 2DFP to detect keypoints in image planes. Subsequently, we project these keypoints along with the predicted scores to 3D space by camera parameters and pixel depths, and use a score-weighted mean-shift clustering to form cluster centers $\mathcal{C} = \{\boldsymbol{C}_j\}$. Specifically, we first initialize K centers by farthest point sampling (FPS) [40], and compute the shifted center positions via the score weighted mean of the neighboring point positions within a radius of $r_c = 0.05$ m. We iterate the above center-shifting 10 times, which is enough to make center positions converge. Finally, to filter out dangling clusters, we also perform a non-maximum suppression with a radius of 0.15 m to remove centers too close to each other and only preserve the centers with more than 5 neighboring points nearby. These cluster centers will be back-projected to generate supervision signals for 2DFP. As training goes on, 2D detection and 3D cluster centers will gradually converge, which yields highly repeatable keypoint detection and maintains a multi-view coherence descriptor learning.

Score map learning. To encourage the network to predict highly repeatable keypoints across multiple views, we use the projections of these cluster centers to supervise the detection. Specifically, we project $\mathcal{C}$ to 2D views, and obtain visible projections $\{\boldsymbol{c}_{ij}\}$ by removing the projections whose z-axis is far from the sensor depth. We generate ground-truth score maps by setting 1 inside a filled circle around projections, and use a binary cross-entropy loss to supervise the score maps $\{\boldsymbol{s}_i\}$ on all the $\mathcal{M}$ frames, as

$$L_{\text{2d\_score}} = -\frac{1}{\mathcal{M}} \sum_{i=1}^{\mathcal{M}} \boldsymbol{m}_i \left[(1-\kappa) \cdot \boldsymbol{s}_{gt_i} \cdot \log \boldsymbol{s}_i + \kappa \cdot (1 - \boldsymbol{s}_{gt_i}) \cdot \log (1 - \boldsymbol{s}_i) \right], \tag{1}$$

where $\boldsymbol{s}_{gt_i}$ is the ground-truth score maps, and $\boldsymbol{m}_i$ is the covisible mask based on the fragment pair overlapping (or mutually observed) areas. Specifically, m_i is constructed by gathering the overlapped points from the lifted depth frames, and then back-projecting them into 2D image planes. By using m_i,

the network will focus on training the image regions with mutually observed viewpoints. $\kappa = 0.05$ is used to balance the gradient between 0 and 1.

To further improve the detection consistency, i.e., the predicted scores should be consistent across views, we also apply a score consistency loss in a multi-view fashion. This can be considered as a general extension of the two-view version [18, 70]. In detail, we assemble the predicted scores from the same cluster center as $\{\boldsymbol{S}_j\}$, and minimize the variance of scores in each $\boldsymbol{S}_j$. Besides, as a good keypoint detection is thought to have a consistent location prediction, we compute the reprojection errors (RPE) of cluster center projections at the location map, and maximize or minimize the average scores if the related RPE is less or greater than the average RPE. To this end, we design the score consistency loss as

$$L_{\text{2d\_cons}} = \frac{1}{|\mathcal{C}|}\sum_{j}\left[\text{var}(\boldsymbol{S}_j) + \overline{\boldsymbol{S}_j}\cdot(\overline{\boldsymbol{e}_j} - \overline{\boldsymbol{e}})\right]. \tag{2}$$

$\text{var}(\cdot)$ calculates the variance of $\boldsymbol{S}_j$. The reprojection error $\boldsymbol{e}$ is computed as $\boldsymbol{e} = ||\boldsymbol{l}(\boldsymbol{p}_n) - \boldsymbol{p}_n||_2$, where $\boldsymbol{p}_n$ is the cluster's projected location, $\boldsymbol{l}(\cdot)$ denotes the sub-pixel level coordinate obtained from the location map when given a rounded projected pixel location. $\overline{\boldsymbol{S}_j}$ and $\overline{\boldsymbol{e}_j}$ are the average scores and reprojection errors of a specific cluster center $\boldsymbol{C}_j$. And $\overline{\boldsymbol{e}}$ is the overall average reprojection error in the current fragment pair.

Location map learning. We enforce the predicted location map to be consistent by minimizing all the reprojection errors of $|\mathcal{C}|$ centers projected on $\mathcal{M}$ image views, with the reprojection error loss defined as follows:

$$L_{\text{2d\_rpe}} = \frac{1}{|\mathcal{C}|\cdot\mathcal{M}}\sum_{n}\boldsymbol{e}_n^2. \tag{3}$$

Note that we aim at enforcing accurate local offsets for location maps, so we only consider the reprojection error between the accurate and rounded projections of the clusters, and omit the original lifted 2D keypoints.

Descriptor map learning. We follow Revaud et al.'s work [16], and employ AP-Loss [72] for contrastive learning, which optimizes the differential approximation of average precision (AP). Instead of sampling anchors on a regular grid in R2D2 [16], we use cluster center projections as anchors to generate descriptor sets, and treat the anchors from the same cluster center as having the same label. The descriptor loss is defined as follows:

$$L_{\text{2d\_desc}} = \frac{1}{|\mathcal{C}|\cdot\mathcal{M}}\sum_{n}L_{\text{AP}}(q_n),\quad L_{\text{AP}}(q) = 1 - \text{AP}(q), \tag{4}$$

where $L_{\text{AP}}(q)$ computes the AP for each query q [72] (i.e., projected cluster centers on image views).

The overall training loss for 2DFP is the weighted sum of the above losses:

$$L_{\text{2d}} = w_{\text{2d\_score}}L_{\text{2d\_score}} + w_{\text{2d\_cons}}L_{\text{2d\_cons}} + w_{\text{2d\_rpe}}L_{\text{2d\_rpe}} + w_{\text{2d\_desc}}L_{\text{2d\_desc}}. \tag{5}$$

We empirically set $w_{\text{2d\_score}} = 10.0$, $w_{\text{2d\_cons}} = 0.1$, $w_{\text{2d\_rpe}} = 0.1$ and $w_{\text{2d\_desc}} = 5.0$.

3.3 3D hybrid feature fusion

Previous studies [19, 20, 22, 23, 66] usually concatenate 2D and 3D features through the pixel-wise association of image and pixel depth. However, in multi-view scenarios, one point in 3D space may correspond to multiple 2D observations, which is regarded as a one-to-many mapping. This also perplexes some view-pooling methods [51, 53, 73] which use fixed number of views. Besides, due to the noise of depth sensors and the estimation error of camera poses, directly binding 2D features to 3D by point locations may also introduce false associations. To this end, we propose a novel loosely coupled 2D-3D feature fusion mechanism, which naturally handles the association issue.

3D soft feature fusion. As illustrated in Figures 1 and 5, we first generate a hybrid point cloud by mixing up K lifted multi-view 2D features (128-dimensional) and N (32-dimensional) 3D features in the same coordinate. To unify the feature-length and avoid feature confusion, we apply feature padding with zero vectors on both types of features, which yields $K + N$ multi-modal features with a length of 160. After that, we perform a soft feature fusion with SA (set abstraction) and FP (feature propagation) layers [40] to extract 3D hybrid features (64-dimensional) from the mixture of $K + N$ features, where the neighboring relationships such as relative positions of multi-modal features are all considered and encoded by the network. In this way, we let the network learn how to deal with the position noise of multi-view observations and 3D points, and gracefully fuse arbitrary numbers of observations with point

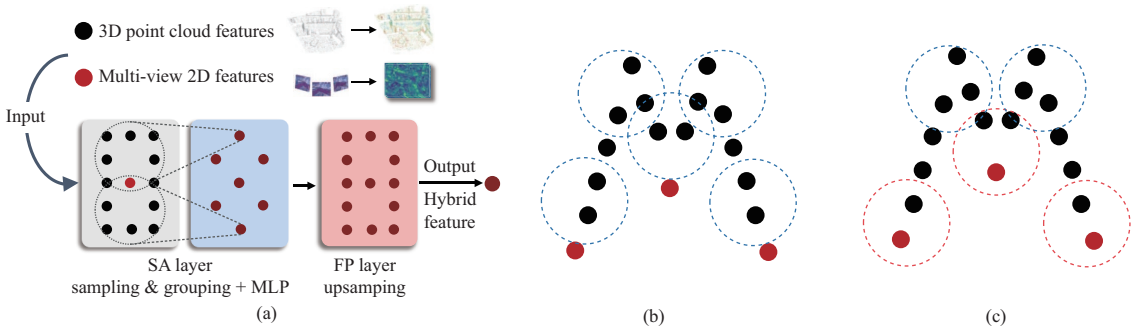

Figure 5 (Color online) A 2D toy example of soft feature fusion. (a) We use the set abstraction and feature propagating layer to fuse and extract hybrid features from the mixture of 2D and 3D features. (b) During feature grouping, 3D point cloud features (black dots) and multi-view 2D features (red dots) have different spatial densities. The default grouping operations of set abstraction [40] might inevitably ignore the sparse 2D features. (c) With the balanced fusion strategy, 2D features would be preferentially grouped.

features at a single 3D location, instead of manually concatenating 2D-3D features as input. Please refer to the supplementary for more details.

Balanced fusion strategy. However, due to the different spatial densities of multi-modal features, the vanilla grouping and encoding operations in set abstraction layers are more likely to exploit the point cloud features and accidentally ignore the multi-view 2D features, as illustrated in Figure 5. To address this issue, we integrate a balanced fusion strategy into the fusion pipeline, which alters and balances the sampling and grouping operations for features from different resources. Specifically, in the sampling step, we separately sample $N_{3d} = 5120$ anchors in pointcloud and $N_{2d} = 1024$ anchors in the 3D positions of the lifted 2D features. The overall anchor is a combination of these two groups of anchors. In the grouping step, we set the number of features to query in the neighborhood of an anchor as $S = 32$, and also set a balancing ratio $\mu = 0.7$. For each grouping operation in the anchor, we search for all the features within a radius of $r = 0.1$. Assuming we find out M multi-view 2D features nearby, when $M \geqslant \mu S$, which means sufficient multi-view 2D features around the anchor, therefore we will preserve μS 2D features in the grouped features. Otherwise, when $M < \mu S$, that is insufficient multi-view 2D features in radius, and we retain all the 2D features found in the grouping operation. As shown in our experiments (Subsection 4.4), this strategy further improves the effectiveness of feature fusion.

3D hybrid feature learning. To train the 3D hybrid features, we adopt the hardest-contrastive loss [44, 74, 75] by mining hard-negative quadruplet pairs. Given a partially overlapping fragment pair, we first sample anchors from the overlapping areas of the point cloud pair to construct positive descriptor pairs $\mathcal{P} = \{(\boldsymbol{f}_i, \boldsymbol{f}_i')\}$. The negative pairs are constructed by mining hardest-negative both for $\boldsymbol{f}_i$ and $\boldsymbol{f}_i'$ in the random subset of the fragment pair, with a safe radius bound to avoid choosing negative pairs too close to each other. Thus, our 3D descriptor loss is defined as

$$L_{\text{3d\_desc}} = \frac{1}{|\mathcal{P}|} \sum_i \left\{ \left[D_{\text{pos}_i} - m_{\text{pos}} \right]_+^2 + \left(\left[m_{\text{neg}} - D_{\text{neg}_i} \right]_+^2 + \left[m_{\text{neg}} - D_{\text{neg}_i}' \right]_+^2 \right) / 2 \right\}, \tag{6}$$

where D_{pos_i} is the Euclidean distance of positive pairs. D_{neg_i} and D_{neg_i}' are the Euclidean distance of negative pairs both for $\boldsymbol{f}_i$ and $\boldsymbol{f}_i'$. $[\cdot]_+$ denotes the maximum operation, as $[\cdot]_+ = \max(\cdot, 0)$. $m_{\text{pos}} = 0.1$ and $m_{\text{neg}} = 1.4$ are the margins for positive and negative pairs.

3.4 3D scoring module

To improve the registration performance when only a few keypoints are allowed, we design a compact 3D scoring module, which aims to find the most repeatable and distinguishable keypoints for our multi-modal features. Unlike previous methods [11, 67] that rely on local value maximum of descriptors or rely on pairwise input for cross-attention [24, 25], we directly take as standalone input descriptors and regress the score without any extra information. During inference, we apply a 3D non-maximum suppression to avoid selecting keypoints too close to each other.

Self-supervised detection score learning. In D3Feat [11], Bai et al. used on-the-fly feature matching results to guide the score prediction. However, at the early stage of network training, the matching results are unstable, which may confuse the score learning with the opposite gradient direction. In contrast, we design a new detector loss that leverages descriptor distinctiveness via statistics of average negative distances D_i^*. We reuse information from 3D descriptor learning, and obtain D_i^* by calculating

the average descriptor distance from anchors to all the negatives. We define our 3D score distinctive loss as

$$L_{\text{3d\_score}} = \frac{1}{|\mathcal{P}|} \sum_i \left[\left(\overline{D^*} - D_i^* \right) \left(\boldsymbol{s}_{\text{3D}i} + \boldsymbol{s}_{\text{3D}i}' \right) \right], \tag{7}$$

where $\overline{D^*}$ is the mean value of average negative distances for each fragment pair. $\boldsymbol{s}_{\text{3D}i}$ and $\boldsymbol{s}_{\text{3D}i}'$ are the predicted scores for anchors from positive pairs. Intuitively, if an anchor comes with a relatively larger average negative distance, the distinctiveness of the associated descriptor is also better, so we will encourage the network to predict a higher score and vice versa. Compared to D3Feat, our 3D score distinctive loss provides consistent gradient direction even at the early stage of training, which benefits the detection performance as shown in our experiment.

Besides, to ensure a better convergence, we also apply 3D score peakiness loss and 3D score consistency. The 3D score peakiness loss aims to maximize the local peakiness of 3D scores by encouraging the local maximal scores to be larger than the local average scores, as

$$L_{\text{3d\_peaky}} = \frac{1}{|\mathcal{P}|} \sum_i [\overline{S_N(\boldsymbol{p}_i)} - \max(S_N(\boldsymbol{p}_i)) + m_{\text{score}}]_+, \tag{8}$$

where $S_N(\boldsymbol{p})$ is the scores of the neighboring points around the point $\boldsymbol{p}$ (within a radius of 0.075 m), and $\overline{S}$ computes the average scores in S. $m_{\text{score}} = 0.3$ is the margin of score peakiness. The 3D score consistency loss is used to maintain the scoring consistency among different fragments, as

$$L_{\text{3d\_cons}} = \frac{1}{|\mathcal{P}|} \sum_i \| \boldsymbol{s}_{\text{3D}i} - \boldsymbol{s}_{\text{3D}i}' \|_2, \tag{9}$$

where $\boldsymbol{s}_{\text{3D}i}$ and $\boldsymbol{s}_{\text{3D}i}'$ are the scores of positive anchors in $\mathcal{P}$. We will show the effectiveness of all these losses in our experiments (Subsection. 4.4).

3.5 Overall training loss

The overall training loss for 3D detection and description training is the weighted sum of these four losses:

$$L_{\text{3d}} = L_{\text{3d\_score}} + L_{\text{3d\_cons}} + L_{\text{3d\_desc}} + w_{\text{3d\_peaky}} L_{\text{3d\_peaky}}. \tag{10}$$

During training, since a large peaky loss might affect the training process by strongly suppressing other informative keypoints and only allowing one point in the preset domain to be activated, so we empirically set $w_{\text{3d\_peak}} = 0.05$.

Please refer to the supplementary material for more details on model structure and training configurations.

4 Experiments

In this section, we first introduce the two datasets we experiment on and the evaluation metrics. Then, we compare our Hyprid3D features with other state-of-the-art methods and perform ablation studies to inspect the design of our modules and losses.

4.1 Dataset preparation

3DMatch [32] is constructed from several RGB-D reconstructions, which contains images and depth with ground-truth poses. We use the official split for training and testing. Since the original evaluation data do not contain the associated image sequence indices, we regenerate point cloud fragment pairs with 3DMatch toolbox [32]. Specifically, we utilize the official 3DMatch toolbox's [32] TSDF fusion to build up point cloud fragment pairs with more than 30% overlap, where each fragment is fused from 50 consecutive depth frames. Since the original RGB and depth frames in 7-Scenes (one subset of 3DMatch) have not been aligned at the pixel level, we warp the depth frames to the RGB image planes according to the extrinsic between RGB and depth sensor, which is similar to that in [76]. Besides, we also adaptively adjust the TSDF volume according to the bounds of view-frustum points (i.e., initialize volume bounds from the convex hull of the lifted depth points), so as to obtain more complete point clouds and also reduce the case that multi-view 2D features are lifted to somewhere without 3D points. To make a fair comparison, we use our generated fragments for all the tests, so the final evaluation results are different from the previous [11, 43, 44].

Redwood [14] contains five indoor RGB-D scanned scenes (apartment, bedroom, boardroom, lobby, and loft) with camera poses. To analyze the generalization of our methods, we only train our model on the 3DMatch dataset and test it on the Redwood dataset. We adopt the same pipeline as 3DMatch to process the data, and randomly sample 200 fragment pairs per scene for evaluation. Different from the previous studies [26, 32], we use the real-world scanned Redwood sequences [14] instead of the synthetic set [77] to show the performance in practice scenarios. Similar to 3DMatch, a point cloud fragment in the datasets is fused by 50 consecutive depth frames with TSDF fusion.

4.2 Evaluation metrics

Keypoint detection evaluation. Following previous studies [11, 26], we adopt the relative repeatability to measure the performance of a keypoint detector. Given the ground truth relative pose of two pointcloud fragments, we can transform a keypoint from the source fragment to a location in the target fragment. A keypoint is repeatable if the distance of the nearest keypoint in the target fragment to its corresponding location is smaller than 10 cm. We calculate the percentage of repeatable keypoints as relative repeatability. As the keypoints are selected by predicted scores, relative repeatability can be regarded as the evaluation metric for 3D keypoint scores.

Feature matching evaluation. Similar to 3DMatch [32], we consider a matched point pair to be inlier if the distance to the correspondence is smaller than $\tau_1 = 10$ cm when given the ground-truth transformation, and use feature matching recall and average inlier match ratio to reveal the descriptor quality for matching. Feature matching recall calculates the percentage of fragment pairs whose inlier match ratio is above $\tau_2 = 5\%$. Average inlier match ratio computes the average inlier match ratio for the fragment pairs whose inlier match ratios are above $\tau_2 = 5\%$.

Point cloud registration evaluation. Registration recall reflects the point cloud registration accuracy using given 3D features, which is calculated by the percentage of successfully aligned pairs with transformation error RMSE < 0.2 m. Following D3Feat [11], we use RANSAC with a maximum of 50000 iterations to estimate the transformations.

In addition, we both evaluate description and detection under different keypoint number settings (i.e., from 50 to 5000), so as to show the extreme performance when only given a small number of keypoints.

4.3 Comparison with the state-of-the-arts

We compare our method with the state-of-the-arts 3D point cloud detection and description methods, including 3DSmoothNet [43], D3Feat [11], FCGF [44], OverlapPredator [24], BYOC [55], and USIP [26]. Among them, BYOC is also supervised with RGB-D data, while the others only use pure geometric features. In addition to that, we compare our method with 2D image features (i.e., SuperPoint [17] and KP2D [18]) since our method leverages image information. To evaluate the performance in point cloud registration, we use 16 associated color images as network input, and lift their detected 2D features and corresponding scores to 3D space via pixel depth and camera parameters. For a fair comparison with 2D image features, we provide the results of Our 2D (RGB), which directly use keypoints predicted by our 2DFP and only accept RGB images as inputs.

Comparisons of keypoint detection. We first compare the relative repeatability in Figures 6 and 7. The Random strategy can be regarded as the lower bound of relative repeatability. As shown in Figure 6, the keypoint provided by USIP [26] do not work well in indoor scenes (Figure 6). The follow-up work D3Feat significantly improves the keypoint detection quality on the noisy indoor scans, but there is still room for improvement. On both 3DMatch and Redwood datasets, our detector achieves the overall best performance and improves the repeatability by a large margin when the keypoint numbers are extremely small (i.e., 128 or less). Besides, the results of KP2D are also very competitive, which highlights the potential value of visual cues for better keypoint selection. We also visualize 2D score maps predicted by KP2D and our 2DFP module in Figure 8, which demonstrates robust keypoint selection ability of our method due to the clustering and projection strategy.

Comparisons of feature matching. We then compare the feature matching performance method with 3D features (3DSmoothNet [43], D3Feat [11], BYOC [55], OverlapPredator [24] FCGF [44]) and 2D features (SuperPoint [17] and KP2D [18]). As shown in Tables 1 and 2 (best results in bold), our complete model (Our 3D) outperforms other methods in all metrics thanks to the effective feature fusion of point cloud and multi-view images. Interestingly, the 2D features (SuperPoint and KP2D) also present reasonable performance even though they are not designed to work in 3D. As RGB images provide more

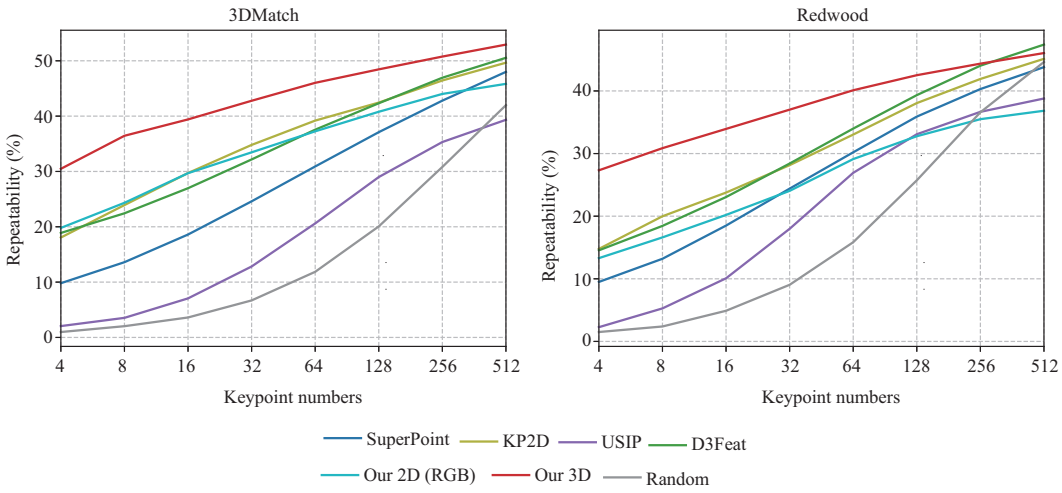

Figure 6 (Color online) Relative repeatability on 3DMatch and Redwood.

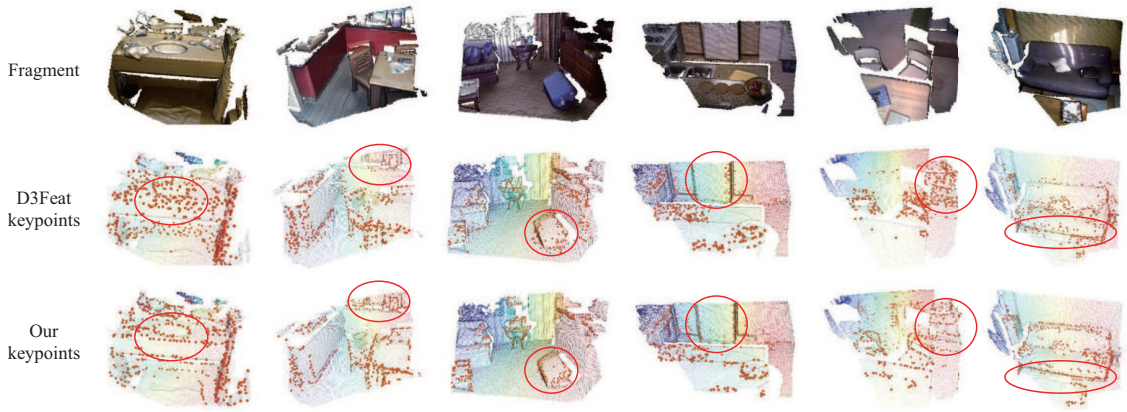

Figure 7 (Color online) Visualization of 3D keypoints (top 500 in the order of scores). We show the 3D keypoints selected by D3Feat [11] and Hybrid3D on the the Redwood dataset [14] and the 3DMatch testing dataset [32]. Thanks to the 3D scoring module and the novel score loss, our Hybrid3D tends to select more keypoints on distinctive areas (e.g., corners or edges), which brings about high relative repeatability, and is the key to achieving leading geometric registration performance when only given limited keypoints.

information than 3D points, they can even achieve higher feature matching recall and smaller average inlier match ratio on the 3DMatch dataset than the 3D point features 3DSmoothNet and FCGF when we select less than 500 keypoints. These results reveal the importance of image information. Besides, Our 2D (RGB), which is our image-only version, surpasses Superpoint and KP2D on feature matching recall and average inlier match ratio, which demonstrates that our cluster-and-projection learning strategy successfully guides the features to be more coherent in 3D space.

Comparisons of point cloud registration. We compare the holistic performance of point cloud registration in Table 3 and visualize in Figure 9. On the track of 2D features, even though KP2D and SuperPoint present convincing performance in feature matching evaluation, they are not directly applicable in point cloud registration. As shown in Table 3 (best results in bold), our full model (Our 3D) achieves SOTA registration performance even without the pairwise point cloud cross-attention as OverlapPredator [24] does. We can surpass many methods that use 5000 keypoints by using only 500 keypoints, which shows the extraordinary robustness of our method. Moreover, when using 50 keypoints, our hybrid3D exceeds the secondary runner (D3Feat) by 22.4% on the Redwood dataset and 26.1% on the 3DMatch dataset.

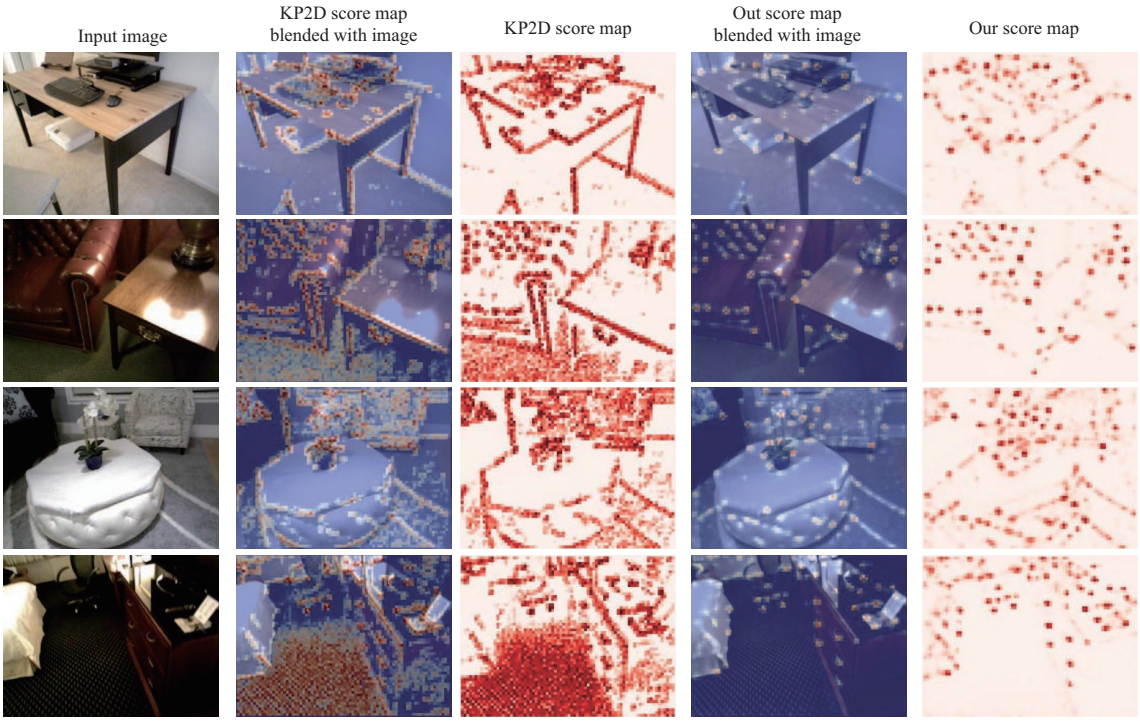

Figure 8 (Color online) Visualization of 2D score maps. We show the score maps predicted by KP2D [18] and our 2DFP module (image-only version) on the the Redwood dataset [14] and the 3DMatch testing dataset [32]. The flavors of different detectors are mainly due to the different training strategies. KP2D uses pairwise supervision with a homography adaptation mechanism, which limits the detection consistency of view-point variations, and also makes the network sensitive to noise and locally repetitive patterns (e.g., noisy detection on the carpet, as shown in the last row). By contrast, our method exploits multi-view supervision with clustering and projection strategy, which yields a more robust keypoint selection at 'middle-level' repeatable patterns (e.g., junctions of furniture).

Table 1 Feature matching evaluation results on 3DMatch dataset

# Keypoints	Feature matching recall (%)							Average inlier match ratio (%)						
	50	100	250	500	1000	2500	5000	50	100	250	500	1000	2500	5000
SuperPoint	75.5	78.3	84.4	86.1	87.3	88.8	89.2	34.4	37.3	39.7	42.1	44.2	45.4	45.5
KP2D	**81.0**	85.1	86.4	87.2	87.1	87.7	87.9	44.7	42.8	43.0	43.3	43.1	43.0	43.0
Our 2D (RGB)	79.1	**87.3**	**90.5**	**91.7**	**91.6**	**91.6**	**91.6**	**54.3**	**51.6**	**52.1**	**51.3**	**50.7**	**50.5**	**50.5**
3DSmoothNet	41.7	55.8	69.3	76.9	82.9	87.7	88.6	14.2	15.8	19.4	22.7	26.3	30.9	33.9
D3Feat	86.0	88.5	91.7	92.6	92.7	92.9	93.0	44.7	45.0	43.6	42.3	40.9	39.1	38.8
FCGF	72.6	86.0	94.0	96.0	97.0	97.2	97.3	17.1	22.0	31.1	38.4	44.7	50.6	53.4
BYOC	36.7	37.5	45.0	51.0	53.7	56.8	59.0	10.8	13.2	14.2	14.6	15.8	17.6	18.8
Predator	87.1	92.0	94.3	95.4	94.9	95.5	95.3	17.1	22.0	31.1	38.4	44.7	50.6	53.4
Our 3D	**92.7**	**95.3**	**96.3**	**96.9**	**97.4**	**97.5**	**97.7**	**59.3**	**57.9**	**57.1**	**56.6**	**55.4**	**54.9**	**56.1**

Table 2 Feature matching evaluation results on Redwood dataset

# Keypoints	Feature matching recall (%)							Average inlier match ratio (%)						
	50	100	250	500	1000	2500	5000	50	100	250	500	1000	2500	5000
SuperPoint	**65.6**	67.9	71.5	73.9	76.4	77.5	77.1	29.6	32.5	35.8	36.8	37.8	38.8	39.0
KP2D	64.2	71.3	74.2	74.1	72.9	73.5	73.5	37.6	35.1	34.3	34.4	35.3	34.6	34.9
Our 2D (RGB)	63.6	**73.6**	**77.2**	**78.5**	**79.3**	**79.0**	**79.0**	**46.5**	**42.3**	**41.9**	**41.5**	**41.0**	**41.1**	**41.1**
3DSmoothNet	48.3	62.5	77.2	83.1	84.4	86.9	87.2	15.9	18.2	22.1	25.6	30.2	34.4	37.2
D3Feat	80.4	82.4	87.8	89.2	88.5	88.0	87.8	38.9	40.4	38.8	38.2	37.5	36.9	38.1
FCGF	77.2	84.2	90.6	91.9	92.7	92.4	92.6	19.1	25.1	33.8	39.9	44.4	48.7	50.8
BYOC	38.4	41.4	46.9	51.7	54.7	56.4	57.7	12.1	13.9	15.4	16.1	17.4	19.6	21.1
Predator	85.6	89.8	91.4	91.3	91.1	90.7	91.5	30.8	38.2	44.9	**47.6**	**48.9**	49.1	47.7
Our 3D	**85.8**	**90.7**	**91.7**	**92.1**	**92.9**	**93.5**	**93.8**	**52.8**	**50.4**	**48.6**	46.6	46.4	**49.8**	**52.9**

Table 3 Point cloud registration evaluation results on 3DMatch and Redwood datasets

# Keypoints	Registration recall on 3DMatch (%)							Registration recall on Redwood (%)						
	50	100	250	500	1000	2500	5000	50	100	250	500	1000	2500	5000
SuperPoint	27.3	39.5	55.6	61.0	67.2	72.2	72.0	17.8	31.5	43.3	50.9	56.0	57.9	57.9
KP2D	**37.1**	**51.6**	66.1	70.7	73.7	73.5	72.3	**23.1**	**35.6**	48.4	53.4	55.2	51.2	51.4
Our 2D (RGB)	30.6	46.6	**70.6**	**77.4**	**79.6**	**80.2**	**78.5**	19.6	31.4	**50.7**	**59.2**	**60.7**	**60.5**	**60.4**
3DSmoothNet	6.9	19.3	43.1	54.9	62.9	70.2	72.4	7.6	23.6	46.9	57.2	61.5	64.8	65.9
D3Feat	48.3	66.5	77.9	83.2	85.1	84.8	82.6	38.4	53.6	69.9	73.3	75.3	70.1	68.0
FCGF	16.2	42.2	69.3	78.4	83.6	84.6	85.3	16.7	41.1	63.1	71.2	71.3	71.6	72.7
BYOC	6.1	16.6	35.3	43.6	47.7	49.9	52.3	8.6	17.8	33.1	39.3	43.7	46.1	44.4
Predator	44.5	67.7	82.5	85.2	**87.5**	87.4	**88.3**	49.1	66.8	76.6	**78.1**	**79.2**	**78.7**	77.6
Our 3D	**60.9**	**74.3**	**85.0**	**88.3**	87.3	**87.5**	87.4	**60.1**	**71.4**	**78.8**	77.4	76.5	**78.7**	**78.4**

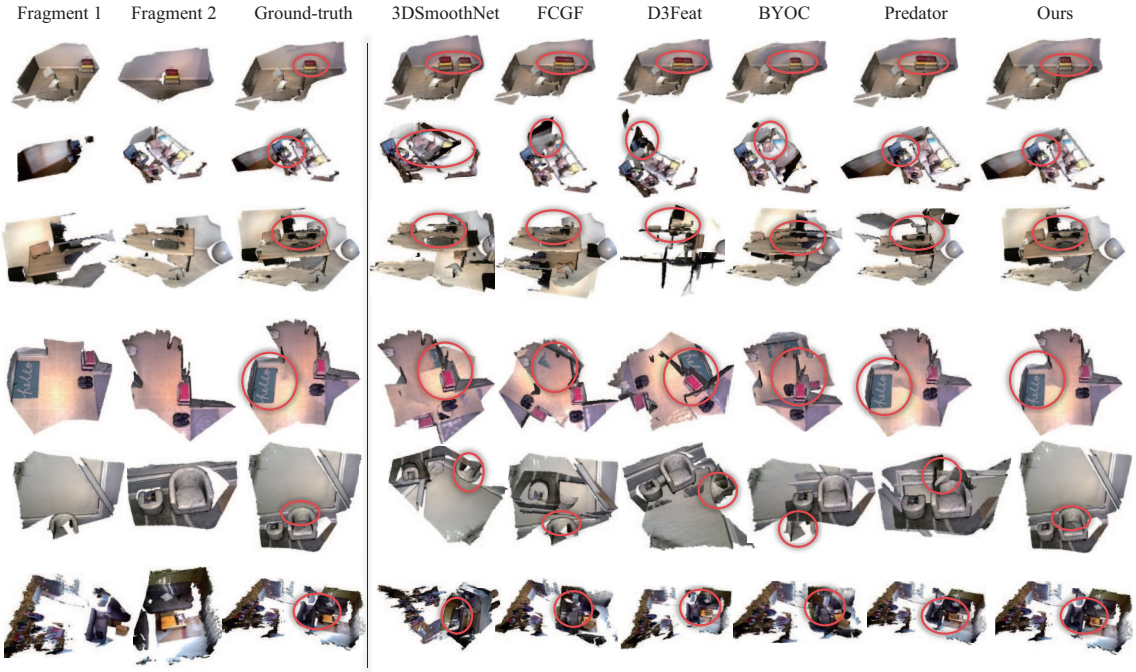

Figure 9 (Color online) Visualization of point cloud registration by different methods with 5000 keypoints and RANSAC. We shows the geometric registration results by 3DSmoothNet (also known as PerfectMatch) [43], FCGF [44], D3Feat [11], BYOC [55], OverlapPredator [24] (also known as Predator) and our Hybrid3D on the the Redwood dataset [14] and the 3DMatch testing dataset [32]. Our method successfully registers fragment pairs with only small overlap areas.

4.4 Ablation study

In this subsection, we perform ablation studies of our network design and strategies on the 3DMatch dataset. We show the results of registration recall and relative repeatability.

2D feature fusion. We ablate the 3DSF and directly use the keypoints from 2DFP to validate the effectiveness of multi-modal feature fusion in 2D image planes. Our 2D (RGB) only accepts RGB images and Our 2D (fuse) uses both RGB images and projected 3D point features. Besides, Our 2D (fuse w/o score cons.) also uses both multi-modal features but is trained without 2D score consistency loss (Eq. (2)). As shown in Table 4 (best results in bold) and Figure 10, Our 2D (fuse) achieves higher registration recall than Our 2D (RGB) because of the favorable 3D point features, and the proposed 2D score consistency loss can further improve the registration performance.

3D feature fusion. We analysis the effectiveness of balanced fusion strategy (Our 3D (w/o balance.)), 3D score consistency loss (Our 3D (w/o cons.)) and 3D score peakiness loss (Our 3D (w/o peak.)). As shown in in Table 4 and Figure 10, the performance without the above strategy or losses will decrease in varying degrees, especially in small keypoint number settings (e.g., 500 keypoints or less).

FCGF with our scoring strategy. To exhibit the generalization of our 3D scoring module and score loss, we train the 3D scoring module on the FCGF, and report the registration recall at FCGF (score).

Table 4 Ablation study: registration recall (%) on 3DMatch

# Keypoints	50	100	250	500	1000	2500	5000
Our 3D	60.9	74.3	85.0	88.3	87.3	87.5	87.4
Our 2D (RGB)	30.6	46.6	70.6	77.4	79.6	80.2	78.5
Our 2D (fuse w/o score cons.)	27.6	48.7	71.7	81.6	84.1	85.0	84.8
Our 2D (fuse)	29.9	51.8	74.7	82.2	85.5	84.6	85.4
Our 3D (w/o balance.)	50.4	66.1	80.4	84.5	87.2	85.7	87.4
Our 3D (w/o cons.)	37.6	60.7	79.4	86.1	86.3	85.7	88.5
Our 3D (w/o peak.)	37.4	52.7	74.2	81.6	84.4	84.3	85.5
Our 3D (rand)	17.8	43.1	70.1	79.1	83.4	84.4	85.3
FCGF	16.2	42.2	69.3	78.4	83.6	84.6	85.3
FCGF (score)	**46.0**	**63.8**	**78.7**	**84.7**	**86.4**	**86.2**	**86.6**
Sparse 5k: FCGF	15.8	36.3	57.9	66.9	69.9	73.0	72.8
Sparse 5k: FCGF (score)	26.4	42.0	61.7	69.1	73.4	74.4	72.8
Sparse 5k: Our 3D	**34.9**	**51.4**	**69.4**	**74.9**	**78.0**	**79.0**	**78.5**
Sparse 1k: FCGF	2.4	4.8	8.0	8.7	10.2	–	–
Sparse 1k: FCGF (score)	2.8	4.6	6.9	8.9	9.9	–	–
Sparse 1k: Our 3D	**8.9**	**16.1**	**26.9**	**33.7**	**52.3**	–	–

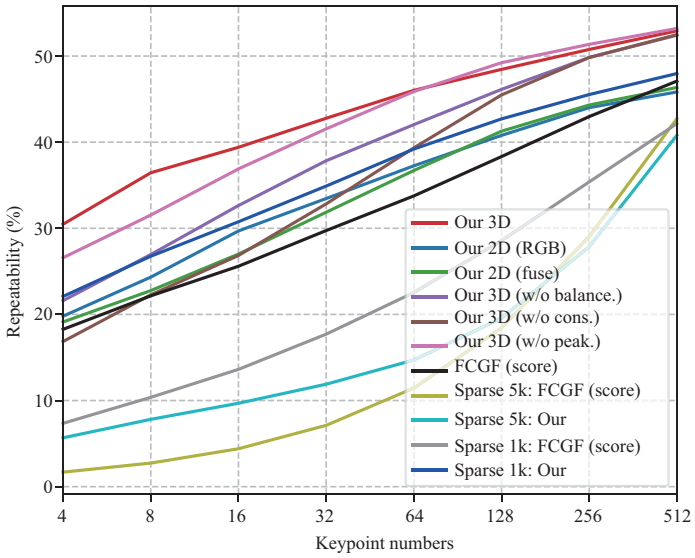

Figure 10 (Color online) Ablation study: relative repeatability on 3DMatch.

The result in Table 4 shows that our scoring strategy significantly improves the FCGF registration performance under small keypoint number settings. However, this result still does not exceed the Our 3D, which shows the vital help of multi-view features.

Sparse point cloud. To further study the registration performance under sparse point cloud settings, we downsample the input point clouds to 5k points and 1k points. We set up the vanilla FCGF and FCGF with our scoring strategy as the baseline. By equipping our scoring strategy to the FCGF, FCGF (score) presents better performance than FCGF in both Table 4 and Figure 10. It means that our score can also work when cooperating with the FCGF features. In Table 4, our full method (Sparse 1k: Our 3D) outperforms FCGF and FCGF (score) when using only 1000 points as input, which shows the great robustness of our Hybrid3D thanks to the effective cross-modal feature fusion.

Please refer to the supplementary material for more experiments on view-pooling methods and scoring strategies.

5 Conclusion

In this study, we propose a novel framework named Hybrid3D, which faithfully exploits and fuses geometry cues from point clouds and texture cues from multi-view images, and then delivers the ability of robust 3D feature detection and description. The experiments demonstrate that the Hybrid3D achieves SOTA registration performance even with very limited keypoints. RGB-D cameras are very common sensors and are widely used in many public datasets and modern devices (e.g., smartphones, tablets, UAVs, and robots). Many other 3D vision tasks, such as 3D scene reconstruction, AR/VR, and SLAM, are also facing the problem of efficiently utilizing multi-view RGB images and point clouds. Extending the proposed framework to these tasks will be our future work. In addition, as we require pixel depth to lift 2D features to 3D space, we only deploy our method to indoor scenarios with commodity-level RGB-D sensors, while the outdoor datasets such as KITTI and RobotCar, or other range sensors such as Laser/LiDAR scanner, are not included (i.e., LiDAR point cloud might not be fully covered with RGB images). Extending our framework to handle outdoor point clouds with partially overlapped images and the discrepancy among different range sensors will be another future work.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (Grant No. 61932003) and ZJU-SenseTime Joint Lab of 3D Vision.

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chen K, Lai Y K, Hu S M. 3D indoor scene modeling from RGB-D data: a survey. Comp Visual Media, 2015, 1: 267–278
2 Chen Y D, Hao C Y, Wu W, et al. Robust dense reconstruction by range merging based on confidence estimation. Sci China Inf Sci, 2016, 59: 092103
3 Chen W, Duan J, Basevi H, et al. PointPoseNet: point pose network for robust 6D object pose estimation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020. 2824–2833
4 Du J, Wang R, Cremers D. DH3D: deep hierarchical 3D descriptors for robust large-scale 6DoF relocalization. In: Proceedings of European Conference on Computer Vision, 2020. 744–762
5 Schneider T, Dymczyk M, Fehr M, et al. Maplab: an open framework for research in visual-inertial mapping and localization. IEEE Robot Autom Lett, 2018, 3: 1418–1425
6 Choy C, Dong W, Koltun V. Deep global registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 2514–2523
7 Lu S, Han J G, Wang L Z, et al. Research on two-stage variable scale three-dimensional point cloud registration algorithm. Laser Optoelectron Prog, 2020, 57: 201503
8 Zhang Z, Dai Y, Sun J. Deep learning based point cloud registration: an overview. Virtual Reality Intell Hardware, 2020, 2: 222–246
9 Guo Y, Bennamoun M, Sohel F A, et al. A comprehensive performance evaluation of 3D local feature descriptors. Int J Comput Vis, 2016, 116: 66–89
10 Stancelova P, Sikudova E, Cernekova Z. Performance evaluation of selected 3D keypoint detector-descriptor combinations. In: Proceedings of International Conference on Computer Vision and Graphics, 2020. 188–200
11 Bai X, Luo Z, Zhou L, et al. D3Feat: joint learning of dense detection and description of 3D local features. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, 2020. 6358–6366
12 Zhang Z, Sun J, Dai Y, et al. VRNet: learning the rectified virtual corresponding points for 3D point cloud registration. IEEE Trans Circuits Syst Video Technol, 2022, 32: 4997–5010
13 Liu B S, Chen X M, Han Y H, et al. Accelerating DNN-based 3D point cloud processing for mobile computing. Sci China Inf Sci, 2019, 62: 212102
14 Park J, Zhou Q, Koltun V. Colored point cloud registration revisited. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), Venice, 2017. 143–152
15 Dusmanu M, Rocco I, Pajdla T, et al. D2-Net: a trainable CNN for joint description and detection of local features. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 8092–8101
16 Revaud J, de Souza C R, Humenberger M, et al. R2D2: reliable and repeatable detector and descriptor. In: Proceedings of Neural Information Processing Systems (NeurIPS), Vancouver, 2019. 12405–12415
17 DeTone D, Malisiewicz T, Rabinovich A. SuperPoint: self-supervised interest point detection and description. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, 2018. 224–236
18 Tang J, Kim H, Guizilini V, et al. Neural outlier rejection for self-supervised keypoint learning. In: Proceedings of the 8th International Conference on Learning Representations (ICLR), 2020
19 Wang C, Xu D, Zhu Y, et al. Densefusion: 6D object pose estimation by iterative dense fusion. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, 2019. 3343–3352
20 Xu D, Anguelov D, Jain A. PointFusion: deep sensor fusion for 3D bounding box estimation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 244–253
21 Chen Y, Yang B, Liang M, et al. Learning joint 2D-3D representations for depth completion. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 10022–10031
22 He Y, Sun W, Huang H, et al. PVN3D: a deep point-wise 3D keypoints voting network for 6DoF pose estimation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, 2020. 11629–11638
23 Qi C, Chen X, Litany O, et al. ImVoteNet: boosting 3D object detection in point clouds with image votes. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, 2020. 4403–4412

24 Huang S, Gojcic Z, Usvyatsov M, et al. PREDATOR: registration of 3D point clouds with low overlap. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 4267–4276
25 Yu H, Li F, Saleh M, et al. CoFiNet: reliable coarse-to-fine correspondences for robust pointcloud registration. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34
26 Li J, Lee G H. USIP: unsupervised stable interest point detection from 3D point clouds. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 361–370
27 Tombari F, Salti S, Stefano L D. Unique signatures of histograms for local surface description. In: Proceedings of European Conference on Computer Vision, 2010. 356–369
28 Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration. In: Proceedings of IEEE International Conference on Robotics and Automation, Kobe, 2009. 3212–3217
29 Rusu R B, Bradski G, Thibaux R, et al. Fast 3D recognition and pose using the viewpoint feature histogram. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010. 2155–2162
30 Aldoma A, Tombari F, Rusu R B, et al. OUR-CVFH—oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation. In: Proceedings of Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium, 2012. 113–122
31 Steder B, Rusu R B, Konolige K, et al. NARF: 3D range image features for object recognition. In: Proceedings of Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ, 2010
32 Zeng A, Song S, Nießner M, et al. 3DMatch: learning local geometric descriptors from RGB-D reconstructions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 199–208
33 Khoury M, Zhou Q, Koltun V. Learning compact geometric features. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), 2017. 153–161
34 Deng H, Birdal T, Ilic S. PPFNet: global context aware local features for robust 3D point matching. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 195–205
35 Deng H, Birdal T, Ilic S. PPF-FoldNet: unsupervised learning of rotation invariant 3D local descriptors. In: Proceedings of European Conference on Computer Vision (ECCV), 2018. 602–618
36 Deng H, Birdal T, Ilic S. 3D local features for direct pairwise registration. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, 2019. 3244–3253
37 Wang Y, Solomon J. Deep closest point: learning representations for point cloud registration. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 3523–3532
38 Aoki Y, Goforth H, Srivatsan R A, et al. PointNetLK: robust & efficient point cloud registration using pointnet. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 7163–7172
39 Qi C, Su H, Mo K, et al. PointNet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017. 77–85
40 Qi C, Yi L, Su H, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, 2017. 5099–5108
41 Thomas H, Qi C, Deschaud J-E, et al. KPConv: flexible and deformable convolution for point clouds. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 6410–6419
42 Wang Y, Sun Y, Liu Z, et al. Dynamic graph CNN for learning on point clouds. ACM Trans Graph, 2019, 38: 1–12
43 Gojcic Z, Zhou C, Wegner J D, et al. The perfect match: 3D point cloud matching with smoothed densities. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, 2019. 5545–5554
44 Choy C, Park J, Koltun V. Fully convolutional geometric features. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 8957–8965
45 Liu B, Wang M, Foroosh H, et al. Sparse convolutional neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, 2015. 806–814
46 Ao S, Hu Q, Yang B, et al. SpinNet: learning a general surface descriptor for 3D point cloud registration. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 11753–11762
47 Spezialetti R, Salti S, Stefano L D. Learning an effective equivariant 3D descriptor without supervision. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 6401–6410
48 Chen Z, Yang F, Tao W. DetarNet: decoupling translation and rotation by siamese network for point cloud registration. 2021. ArXiv:2112.14059
49 Bai X, Luo Z, Zhou L, et al. PointDSC: robust point cloud registration using deep spatial consistency. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 15859–15869
50 Sarlin P-E, DeTone D, Malisiewicz T, et al. SuperGlue: learning feature matching with graph neural networks. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 4938–4947
51 Zhou L, Zhu S, Luo Z, et al. Learning and matching multi-view descriptors for registration of point clouds. In: Proceedings of European Conference on Computer Vision, Munich, 2018. 527–544
52 Li L, Zhu S, Fu H, et al. End-to-end learning local multi-view descriptors for 3D point clouds. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, 2020. 1916–1925
53 Huang H, Kalogerakis E, Chaudhuri S, et al. Learning local shape descriptors from part correspondences with multiview convolutional networks. ACM Trans Graph, 2018, 37: 1–14
54 Banani M E, Gao L, Johnson J. UnsupervisedR&R: unsupervised point cloud registration via differentiable rendering. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 7129–7139
55 Banani M E, Johnson J. Bootstrap your own correspondences. In: Proceedings of Computer Vision and Pattern Recognition, 2021. 6433–6442
56 Sipiran I, Bustos B. Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes. Vis Comput, 2011, 27: 963–976
57 Zhong Y. Intrinsic shape signatures: a shape descriptor for 3D object recognition. In: Proceedings of IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, 2009. 689–696
58 Yew Z J, Lee G H. 3DFeat-Net: weakly supervised local 3D features for point cloud registration. In: Proceedings of European Conference on Computer Vision, Munich, 2018. 630–646
59 Liang M, Yang B, Wang S, et al. Deep continuous fusion for multi-sensor 3D object detection. In: Proceedings of European Conference on Computer Vision, Munich, 2018. 663–678
60 Yoo J H, Kim Y, Kim J, et al. 3D-CVF: generating joint camera and LIDAR features using cross-view spatial feature fusion for 3D object detection. In: Proceedings of European Conference on Computer Vision, 2020. 720–736

61 Dai A, Nießner M. 3DMV: joint 3D-multi-view prediction for 3D semantic scene segmentation. In: Proceedings of European Conference on Computer Vision, 2018. 452–468
62 Jaritz M, Gu J, Su H. Multi-view pointnet for 3D scene understanding. In: Proceedings of IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019
63 Zhang J, Zhu C, Zheng L C, et al. Fusion-aware point convolution for online semantic 3D scene segmentation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 4534–4543
64 Huang S S, Ma Z Y, Mu T J, et al. Supervoxel convolution for online 3D semantic segmentation. ACM Trans Graph, 2021, 40: 1–15
65 Xiang R, Zheng F, Su H, et al. 3dDepthNet: point cloud guided depth completion network for sparse depth and single color image. 2020. ArXiv:2003.09175
66 Xing X, Cai Y, Lu T, et al. 3DTNet: learning local features using 2D and 3D cues. In: Proceedings of International Conference on 3D Vision (3DV), Verona, 2018. 435–443
67 Wang B, Chen C, Cui Z, et al. P2-Net: joint description and detection of local features for pixel and point matching. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 16004–16013
68 Pham Q-H, Uy M A, Hua B-S, et al. LCD: learned cross-domain descriptors for 2D-3D matching. In: Proceedings of Computer Vision and Pattern Recognition, New York, 2020. 11856–11864
69 Feng M, Hu S, Ang M H, et al. 2D3D-MatchNet: learning to match keypoints across 2D image and 3D point cloud. In: Proceedings of International Conference on Robotics and Automation (ICRA), 2019. 4790–4796
70 Christiansen P H, Kragh M F, Brodskiy Y, et al. UnsuperPoint: end-to-end unsupervised interest point detector and descriptor. 2019. ArXiv:1907.04011
71 Lindenberger P, Sarlin P-E, Larsson V, et al. Pixel-perfect structure-from-motion with featuremetric refinement. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 5987–5997
72 He K, Lu Y, Sclaroff S. Local descriptors optimized for average precision. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 596–605
73 Wang C, Pelillo M, Siddiqi K. Dominant set clustering and pooling for multi-view 3D object recognition. In: Proceedings of British Machine Vision Conference, London, 2017
74 Mishchuk A, Mishkin D, Radenovic F, et al. Working hard to know your neighbor's margins: local descriptor learning loss. In: Proceedings of Computer Vision and Pattern Recognition, Long Beach, 2017. 4826–4837
75 Law M T, Thome N, Cord M. Quadruplet-wise image similarity learning. In: Proceedings of IEEE International Conference on Computer Vision, Sydney, 2013. 249–256
76 Brachmann E, Rother C. Learning less is more-6D camera localization via 3D surface regression. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 4654–4662
77 Choi S, Zhou Q, Koltun V. Robust reconstruction of indoor scenes. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, 2015. 5556–5565