Bootstrap Your Own Correspondences

Mohamed El Banani   Justin Johnson
University of Michigan
{mbanani, justincj}@umich.edu

Abstract

Geometric feature extraction is a crucial component of point cloud registration pipelines. Recent work has demonstrated how supervised learning can be leveraged to learn better and more compact 3D features. However, those approaches' reliance on ground-truth annotation limits their scalability. We propose BYOC: a self-supervised approach that learns visual and geometric features from RGB-D video without relying on ground-truth pose or correspondence. Our key observation is that randomly-initialized CNNs readily provide us with good correspondences; allowing us to bootstrap the learning of both visual and geometric features. Our approach combines classic ideas from point cloud registration with more recent representation learning approaches. We evaluate our approach on indoor scene datasets and find that our method outperforms traditional and learned descriptors, while being competitive with current state-of-the-art supervised approaches.

1. Introduction

One's ability to align two views of the same scene is closely intertwined with their ability to identify corresponding points between the two views. The duality between correspondence estimation and point cloud registration has long been recognized and serves as the basis for many approaches in both problems . Given an accurate registration of a scene, one can easily extract correspondences between the two views. Conversely, given point correspondences, one can easily register two views of a scene. *Can we leverage this cycle to jointly learn both correspondence estimation and point cloud registration from scratch?*

At the core of this cycle is the ability to generate good feature descriptors for points in the scene. The prevailing approach to 3D feature learning relies on preregistered scenes to sample ground-truth correspondences for the supervised training of a feature encoder. This is done by sampling positive and negative feature pairs and applying triplet [12, 32, 35, 57] or contrastive [3, 12, 56] losses. While very successful, these approaches require us to have
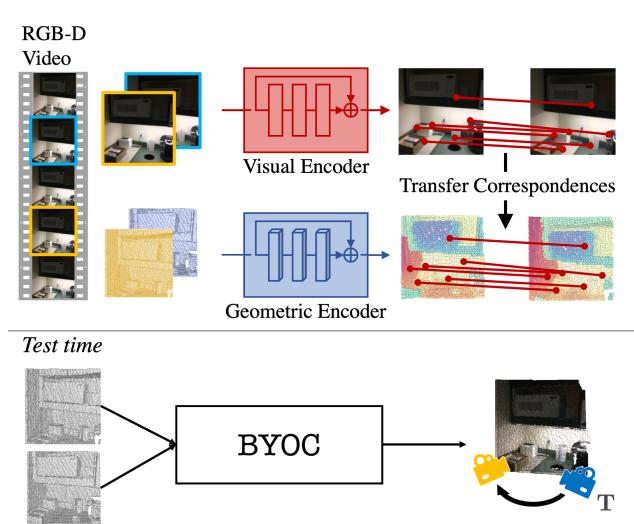

Figure 1. BYOC estimates visual correspondences and uses them to train a visual and a geometric encoder on RGB-D video frames. At test time, it can successfully register raw point clouds.

already registered the raw depth or RGB-D scans to generate the training data. This limits this approach to data that can be successfully registered with automated approaches like COLMAP [46]. Ideally, we would leverage the success of supervised approaches without relying on ground-truth correspondence labels.

To this end, we propose Bootstrap Your Own Correspondences (BYOC): a self-supervised end-to-end approach that learns point cloud registration by leveraging pseudo-correspondence labels. Our approach extracts pseudo-correspondences using the features of a randomly initialized feature encoder. We use the sampled correspondences to register the point clouds and apply losses based on the quality of the registration to train the feature encoders. This allows us to slowly bootstrap[1] the feature learning process and learn from RGB-D scans without relying on any pose or correspondence supervision.

---

[1]We use *bootstrap* in its idiomatic rather than its statistical sense.

This approach works well for registering RGB-D frames, but it is less effective for raw point clouds. This is primarily due to the fact that randomly initialized 2D CNNs produce more distinctive features than current point cloud encoders, as shown in Fig. 3. We leverage this observation and propose bootstrapping the geometric feature learning using visual correspondences. We do this by using the estimated visual correspondences, as opposed to ground-truth correspondences [3, 12, 32, 35, 56, 57], to train the geometric encoder. We train the geometric encoder by adapting SimSiam [8], a non-contrastive self-supervised approach, for 3D representation learning. Unlike typical contrastive self-supervised approaches, SimSiam allows us to train the model using only positive pairs without requiring negative sampling or momentum encoders.

Our work draws inspiration from two sources: iterative closest point algorithm (ICP) [4, 9, 61] and self-supervised learning with pseudo-labels [7, 26, 34]. While seemingly different, the same intuition lies at the core of both lines of work. ICP is a registration algorithm that assumes that the closest points between two point clouds correspond to each other. Through iterative refinement and resampling, it can register roughly aligned point clouds. Meanwhile, self-supervised learning with pseudo-labels learns to predict pseudo-labels in the form of the current top prediction [34], feature clusters [7], or even a previous prediction [26]. Through redefining the labels over time, the model can progressively learn better representations. Both rely on the observation that pseudo-labels in a well-structured space (*i.e.*, similar entities already lie close to each other) can provide a valuable learning signal. This is particularly relevant for learning due to the observation that CNNs, *even* when randomly initialized, are good feature extractors [42, 51].

We evaluate our approach on two indoor scene datasets: ScanNet [13] and 3D Match [60]. Despite the simplicity of our approach, it outperforms hand-crafted features as well as several supervised baselines, while being competitive with current state-of-the-art supervised approaches.

In summary, we propose a self-supervised approach that uses sampled correspondences from randomly-initialized feature encoders to learn point-wise features for point cloud registration (Sec. 3.1). We further demonstrate how visual correspondences can further improve geometric feature learning (Sec. 3.2). We demonstrate the efficacy of this approach on point cloud registration (Sec. 4.1) and correspondence estimation (Sec. 4.2).

2. Related Work

3D Feature Descriptors. Early work on feature point extraction can be traced back to using corners for stereo matching [38]. The core intuition of extracting features based on histograms of gradients was later extended to 3D features [29, 30, 44, 45, 49]. More recently, the focus has

shifted towards leveraging supervised learning for 3D feature learning [3, 12, 14–16, 22, 32, 35, 55, 57, 64]. The common approach is to sample positive and negative pairs between two views and then use them in triplet [12, 32, 35, 57] or contrastive [3, 12, 15, 56] losses. Other approaches propose applying unsupervised learning on reconstructed scenes [14, 56, 64]. While those approaches do not explicitly use ground-truth pose, they rely on reconstructed scenes which are generated using ground-truth pose. Unlike prior work, our approach learns directly from RGB-D scans without relying on ground-truth pose, and focuses on point cloud registration as an end task.

Point Cloud Registration. Early work on point cloud registration assumed perfect correspondence between the point clouds [2, 36]. This assumption was later relaxed by ICP by assuming that the closest point is the correspondence [4, 9, 61]. While this assumption holds for several applications (*e.g.*, registering scans from a high frame-rate scanner or fine-tuning alignment), it is challenged by large transformations and partially overlapping point clouds. Later work focused on designing feature descriptors for establishing correspondence and using robust estimators such as RANSAC to handle noise and outliers [50, 62]. For a review, see [39]. This has been extended further by incorporating learning into the registration process [5, 6, 10, 19, 21, 28, 40, 58]. Finally, recent work has proposed self-supervised approaches for registering objects [1, 27, 28, 54, 55, 58, 59] or reconstructed scenes [14, 32, 64]. Those approaches operate on dense point clouds that are constructed from aligned partial views. Hence, while the method might be self-supervised, the overall approach still requires ground-truth annotation. We are inspired by this line of work and extend it by learning directly from RGB-D scans instead of reconstructed scenes.

Self-supervised learning. Self-supervised learning refers to approaches that apply supervised learning to tasks where the data itself serves as the supervision. This idea has been very popular for 2D representation learning with the goal of learning representations that generalize to downstream tasks [8, 17, 18, 20, 24, 26, 48]. Recently, PointContrast [56] and DepthContrast [63] demonstrated how to extend this formulation to 3D representation learning. We are inspired by this line of work but differ from it in several ways. First, our goal is to learn good features for registration, not for different downstream tasks. Second, we learn from RGB-D videos, not reconstructed scenes like [56]. Also, we learn point-level representations, not scene-level representations like [63]. Finally, while prior work has focused on using contrastive learning, we show that non-contrastive learning [8, 26] can be very effective for 3D feature learning despite being far simpler.
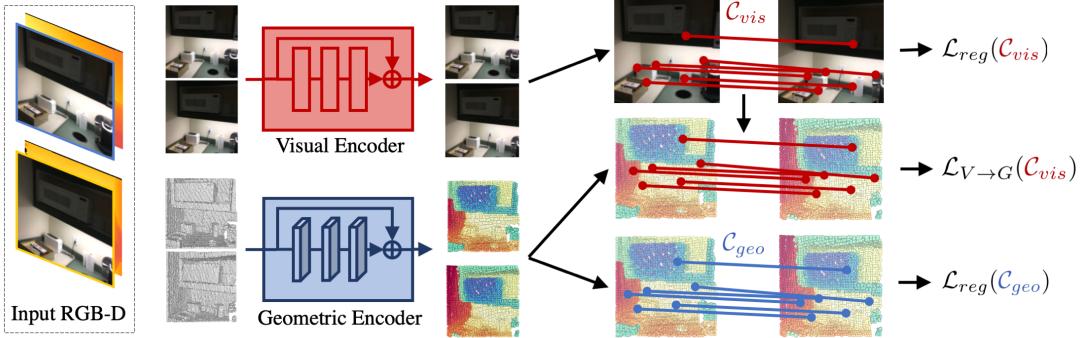
Figure 2. **BYOC**. Our model takes as input two RGB-D images of a scene. First, we extract visual features from the images and geometric features from the point clouds. This results in two point clouds where each point has a 3D location, visual feature, and geometric feature. We then extract correspondences from the visual and geometric features. Those correspondences are used to estimate a transformation and compute a registration loss. We also apply a feature similarity loss on geometric features sampled using the visual correspondences.

3. Approach

The goal of this work is to learn geometric point cloud registration from RGB-D video without relying on pose or correspondence supervision. Our approach, shown in Fig. 2, has three major components: visual registration, geometric registration, and correspondence transfer. The first two components are based on the traditional registration pipeline of feature extraction, correspondence estimation, and geometric fitting. The only difference between them is whether the features are extracted using a visual encoder from the image or a geometric encoder from the point cloud. The third component is based on SimSiam [8] and applies a feature similarity loss on pairs of geometric features that are sampled using visual correspondences. Our key insight is that randomly initialized CNNs produce features that allow for coarse correspondence estimation and registration. This allows us to bootstrap the learning of both visual and geometric encoders by using estimated correspondences with registration and feature similarity losses.

3.1. Point Cloud Registration

Given two point clouds, $\mathcal{P}_0$ and $\mathcal{P}_1$, point cloud registration is the task of finding the transformation $\mathbf{T} \in \mathrm{SE}(3)$ that aligns them. Registration approaches commonly consist of three stages: feature extraction, correspondence estimation, and geometric fitting. In our approach, we register the point cloud pair using either visual or geometric features. Below we discuss each of these steps in detail.

Geometric Feature Extraction. The geometric encoder extracts features based on the geometry of the point cloud. We first generate a point cloud for each view using the input depth and known camera intrinsic matrix. We then encode each point cloud using a sparse 3D convolutional network [11, 25]. We use this network due to its success as a back-end for supervised registration approaches [10, 12, 21]

and 3D representation learning [56, 63]. This network applies sparse convolution to a voxelized point cloud; allowing it to extract features based on local geometry while maintaining a quick run-time. Similar to prior work [12, 56, 63], we find that a voxel size of 2.5 cm works well for indoor scenes. This step maps our input RGB-D image, $I_0, I_1 \in \mathbb{R}^{4 \times H \times W}$ to $\mathcal{P}_0, \mathcal{P}_1 \in \mathbb{R}^{N \times (3+F)}$ where each point cloud has N points, and each point p is represented by a 3D coordinate $\mathbf{x}_p$ and a F-dimensional geometric feature vector $\mathbf{g}_p$.[2] We use a feature dimension of 32.

Visual Feature Extraction. The visual encoder extracts features based on the image. We use a ResNet encoder with two residual blocks as our image encoder and map each pixel to a feature vector of size 32. We use the projected 3D coordinates of the voxelized point cloud from the geometric encoder to index into the 2D feature map. This allows us to generate a point cloud for each input RGB-D image, where each point $p \in \mathcal{P}$ has a 3D coordinate $\mathbf{x}_p$, a visual feature $\mathbf{v}_p$, and a geometric feature $\mathbf{g}_p$. Since each point can be represented by a visual or a geometric feature, we can easily transfer the correspondences between the different feature modalities as shown in Sec. 3.2. We note that we only use the visual encoder during training to bootstrap the geometric feature learning. At test time, we register point clouds using only the geometric encoder.

Correspondence estimation. We estimate the correspondences between the two input views for each feature modality to output two sets of correspondences: $\mathcal{C}_{vis}$ and $\mathcal{C}_{geo}$. We first generate a list of correspondences by finding the nearest neighbor to each point in the appropriate feature space. Since each point cloud has N points, we end up with 2N candidate correspondences for each modality.

---

[2]Voxelization will result in point clouds of varying dimension. We use heterogeneous batching to handle this in our implementation, but assume that point clouds have the same size in our discussion for clarity.

The candidate correspondences will likely contain a lot of false positives due to poor matching, repetitive features, and occluded or non-overlapping portions of the image. The common approach is to filter the correspondences based on some criteria of uniqueness or correctness. Recent approaches propose learning networks that estimate a weight for each correspondence [10, 21, 40]. In this work, we leverage the method proposed by [19] of using a weight based on Lowe's ratio [37]. Given two point clouds, $\mathcal{P}_0$ and $\mathcal{P}_1$, we find the correspondences of point $p \in \mathcal{P}_0$ by finding the two nearest neighbors q_p and q_{p,nn_2} to p in $\mathcal{P}_1$ in feature space. We can calculate the Lowe's ratio weight as follows:

$$w_{p,q_p} = 1 - \frac{D(\mathbf{f}_p, \mathbf{f}_{q_p})}{D(\mathbf{f}_p, \mathbf{f}_{q_{p,nn_2}})} \qquad (1)$$

where D is cosine distance, and $\mathbf{f}_p$ is either the visual or the geometric feature descriptor depending on the feature modality used. It is worth noting that this formulation is similar to the triplet loss often used in contrastive learning, where q_p is the positive sample and q_{p,nn_2} is the hardest negative sample. We use the resulting weights to rank the correspondences and only include the top k correspondences. We use $k = 400$ in our experiments. Each element of our correspondence set $\mathcal{C}$ consists of the two corresponding points and their weight $(p, q, w_{p,q})$.

Geometric Fitting. For each set of correspondences, we estimate the transformation, $\mathbf{T}^* \in \mathrm{SE}(3)$ that would minimize the mean-squared error between the aligned correspondences:

$$E(\mathcal{C}, \mathbf{T}) = \sum_{(p,q_p,w)\in\mathcal{C}} \frac{w}{\sum_{\mathcal{C}} w} \|\mathbf{x}_{q_p} - \mathbf{T}(\mathbf{x}_p)\| \qquad (2)$$

This problem can be reformulated as a weighted Procrustes problem [23, 31, 47, 52] allowing for weights to be integrated into the operation to improve the optimization process while maintaining differentiability with respect to the weights [10]. We adopt this formulation due to its relative simplicity and ease of incorporation within an end-to-end trainable system.

Despite having filtered the correspondences, the correspondence set might still include some outliers that would result in an incorrect geometric fitting. We adopt the randomized optimization used in [19], and similarly find that we get the best performance by only using it at test time.

Registration Loss. Our registration loss is defined with respect to our correspondence set and the estimated transformation as follows:

$$\mathcal{L}_{reg}(\mathcal{C}) = \underset{\mathbf{T}\in\mathrm{SE}(3)}{\arg\min} E(\mathcal{C}, \mathbf{T}) \qquad (3)$$

There are a few interesting things about this loss. First, the gradients are back-propagated to the feature encoder
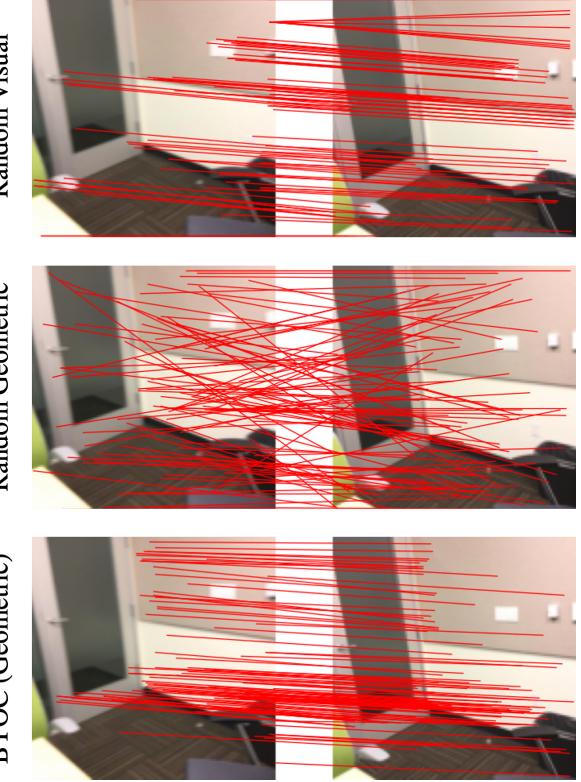

Figure 3. **Randomly-initialized CNNs are good feature extractors.** We can estimate good correspondences from random visual features, but not random geometric features. We leverage this observation to bootstrap the learning of geometric features using visual correspondences.

through both the weights, w, *and* the transformation, $\mathbf{T}$. Hence, the loss can be formulated without using the weights. We find that using the weight improved the performance of visual registration while deteriorating the performance of geometric registration. Therefore, in our model, we only apply the weighting to the visual registration branch while removing it from the geometric branch.

Second, the loss operates as a weighted sum over the residuals. Specifically, the loss is minimized if the correspondence with the lowest residual error has the highest weight. Since the weights are L1 normalized, the relative weighing of the correspondences matters. Removing the normalization results in an obvious degeneracy since the loss can be minimized by driving the weights to 0, which can be achieved by mode collapse. Finally, the weighted loss closely resembles a triplet loss since we estimate both a positive (first nearest neighbor) and a hardest negative (second nearest neighbor) sample. However, unlike the commonly used margin triplet loss, this formulation does not require defining a margin as it operates on the ratio of distances rather than their absolute value.

3.2. Visual → Geometric

The approach outlined in Sec. 3.1 works well with visual features, but it is less effective with geometric features. The reason for this becomes apparent once we consider the registration performance using features from randomly initialized encoders. As shown in Fig. 3, we observe that the features extracted from a randomly initialized visual encoder provide some distinctive output, while a random geometric encoder's outputs are more random. This has a strong impact on registration as shown in Tab. 2.

Ideally, we would leverage the good visual correspondence to further bootstrap the geometric feature learning. We observe that geometric feature learning approaches typically define metric learning losses using sampled correspondences [3, 12, 22, 35, 57]. We adapt this approach to the unsupervised setting by sampling feature pairs using visual correspondences. This is simple in our approach since each point has both a visual feature and a geometric feature, so transferring correspondences is simply indexing into another tensor. Since the correspondences act as indices, the loss is only back-propagated to the geometric encoder.

Current 3D feature learning approaches rely on both positive and negative pairs to define triplet [12, 32, 35, 57] or contrastive [3, 12, 56] losses. However, as noted in the literature, those losses can be difficult to apply due to their susceptibility to mode collapse and sensitivity to hyperparameter choices and negative sampling strategy [12, 56, 63]. Those issues are amplified in our setting since the visual correspondences only provide us with estimated, not ground-truth, positive samples. Instead of the typical contrastive setup, we adapt the recently proposed non-contrastive self-supervised learning approaches [8, 26] to the point cloud setting. We use SimSiam [8] due to its simplicity and strong performance: it does not require negative sampling or a momentum encoder.

We adapt SimSiam by applying it to the geometric features of visually corresponding points instead of different augmentations of the same image. Given a correspondence $(p, q) \in \mathcal{C}_{vis}$, we first project the features using a two-layer MLP projection head and apply a stop-gradient operator on the features:

$$\mathbf{z}_p = project(\mathbf{g}_p). \tag{4}$$

$$\overline{\mathbf{g}_p} = \texttt{stopgradient}(\mathbf{g}_p). \tag{5}$$

We then compute the loss based on the cosine distance between each geometric feature and the projection of its correspondence:

$$\mathcal{L}_{V \to G}(\mathcal{C}_{vis}) = \frac{1}{|\mathcal{C}_{vis}|} \sum_{(p,q) \in \mathcal{C}_{vis}} D(\overline{\mathbf{g}_p}, \mathbf{z}_q) + D(\mathbf{z}_p, \overline{\mathbf{g}_q}) \tag{6}$$

where D is the cosine distance function and $\mathcal{C}_{vis}$ is the set of visual correspondences.

4. Experiments

We evaluate our approach on point cloud registration of indoor scenes. We train our model on ScanNet, a large dataset of indoor scenes, and evaluate it on ScanNet and the 3D Match registration benchmark. Our experiments aim to answer two questions: (1) can we learn accurate point cloud registration from bootstrapped correspondences?; (2) can we leverage RGB-D video at training time to train better geometric encoders?

BYOC variants. We consider two variants of our model: BYOC-Geo and BYOC. BYOC-Geo is trained only on depth pairs using the geometric registration loss. This variant applies the bootstrapping idea without leveraging the visual correspondence. BYOC, shown in Fig. 2, is trained using RGB-D pairs, but only uses the geometric encoder for registration at test time. Since BYOC uses visual correspondences to train the geometric features, we use data augmentation to further improve the geometric feature learning. We sample random rotations and apply them to the point cloud before the geometric encoder. This is a common augmentation in 3D feature learning [12, 56] and is intended to improve the learned feature's rotational equivariance. We note that training BYOC-Geo with rotation augmentation greatly deteriorates its performance.

Datasets. We evaluate our approach on two datasets of indoor scenes: ScanNet [13] and 3D Match [60]. While both datasets provide RGB-D video annotated with ground-truth camera poses, 3D Match provides an additional geometric registration benchmark that is more challenging due to the larger viewpoint changes. ScanNet provides pose annotated RGB-D video for 1513 scenes, while 3D Match's RGB-D video dataset only spans 101 scenes. We emphasize that we only use RGB-D video and camera intrinsics for training our model. We use the official train/valid/test scene split for both datasets, and generate view pairs by sampling image pairs that are 20 frames apart. This results in 1594k/12.6k/26k RGB-D pairs for ScanNet and 122k/1.5k/1.5k RGB-D pairs for 3D Match.

Training Details. We train our model with the Adam [33] optimizer using a learning rate of 10^{-4} and momentum parameters of (0.9, 0.99). We train each model for 200K iterations with a batch size of 8. We implement our models in PyTorch [41], with extensive use of PyTorch3D [41], Open3D [65], and Minkowski Engine [11]. The code is available at https://github.com/mbanani/byoc.

4.1. Point Cloud Registration

We first evaluate our approach on point cloud registration on ScanNet and report our results in Tab. 1. Given two point clouds, we estimate the transformation $\mathbf{T} \in \mathrm{SE}(3)$ that would align the point clouds. We emphasize that we

| | Train Set | Pose Sup. | Rotation | | | | | Translation | | | | | Chamfer | | | | |
| | | | Accuracy ↑ | | | Error ↓ | | Accuracy ↑ | | | Error ↓ | | Accuracy ↑ | | | Error ↓ | |
			5°	10°	45°	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
ICP (Point-to-Point)	-		31.7	55.6	99.6	10.4	8.8	7.5	19.4	74.6	22.4	20.0	8.4	24.7	40.5	32.9	14.1
ICP (Point-to-Plane)	-		54.4	68.0	98.6	8.6	3.6	30.0	36.7	70.4	23.6	18.0	31.6	43.1	53.5	229.5	8.2
FPFH [44] + Weighted Procrustes	-		22.2	48.2	84.9	27.8	10.4	7.4	19.6	56.3	54.1	25.3	17.5	46.8	61.2	26.5	5.8
FPFH [44] + RANSAC	-		34.1	64.0	90.3	20.6	7.2	8.8	26.7	66.8	42.6	18.6	27.0	60.8	73.3	23.3	2.9
FCGF [12] + Weighted Procrustes	3D Match	✓	54.1	73.3	92.2	15.3	4.3	30.8	46.2	73.0	35.0	11.6	45.6	67.4	76.4	21.5	1.4
FCGF [12] + RANSAC	3D Match	✓	75.3	87.7	95.6	9.7	2.5	39.7	64.9	86.5	20.8	6.4	62.5	83.1	88.2	13.0	0.6
FCGF [12] + DGR [10]	3D Match	✓	83.6	90.5	95.2	9.0	1.7	57.6	78.8	91.3	17.1	4.2	76.5	89.4	91.8	10.7	0.3
FCGF [12] + 3D MV Reg [21]	3D Match	✓	87.7	93.2	97.0	6.0	1.2	69.0	83.1	91.8	11.7	2.9	78.9	89.2	91.8	10.2	0.2
BYOC	3D Match		66.5	85.2	97.8	7.4	3.3	30.7	57.6	88.9	16.0	8.2	54.1	82.8	89.5	9.5	0.9
BYOC-Geo	ScanNet		80.3	92.8	98.8	4.8	2.3	46.5	74.6	94.6	10.6	5.4	71.9	91.1	94.5	7.2	0.5
BYOC + RANSAC	ScanNet		81.3	92.8	98.4	5.6	2.4	37.8	69.7	92.1	13.3	6.4	67.7	89.8	93.5	7.7	0.5
BYOC	ScanNet		86.5	95.2	99.1	3.8	1.7	56.4	80.6	96.3	8.7	4.3	78.1	93.9	96.4	5.6	0.3

Table 1. **Pairwise Registration on ScanNet.** We outperform existing registration pipelines that use traditional or learned geometric feature descriptors with a RANSAC or Weighted Procrustes estimator. Furthermore, we perform on-par with supervised approaches that were trained on 3D Match, demonstrating the utility of unsupervised training in this domain. *Pose Sup.* indicates pose supervision.

discard the visual encoder at the test time and only use the geometric encoder on the point cloud input.

Baselines. We compare our approach to both classical hand-crafted and supervised learning approaches. We first compare our approach against two variants of ICP [43]. ICP is an important baseline since it is both an inspiration of this work and a classical point cloud registration algorithm. We also compare against a RANSAC-based aligner using FPFH [44] or FCGF [12] 3D feature descriptors. FPFH [44] is a hand-crafted 3D feature descriptor that represents a point by a histogram of the spatial relationships to its nearest neighbors. FPFH is one of the best non-learned 3D feature descriptors and is representative of the performance of hand-crafted 3D features. FCGF [12] is a recently proposed learned 3D feature descriptor that combines sparse 3D convolutional networks with contrastive losses trained on ground-truth correspondences to achieve state-of-the-art performance on several registration benchmarks.

We also compare against Deep Global Registration [10] and 3D Multiview Registration[3] [22]: two supervised approaches that learn to estimate correspondences on top of FCGF features. Those approaches use supervision for both feature learning and correspondence estimation, while our approach is unsupervised for both.

Evaluation Metrics. We evaluate the pairwise registration by calculating the rotation and translation error between the predicted and ground-truth transformation as follows:

$$E_{\text{rotation}} = \arccos(\frac{Tr(\mathbf{R}_{pr}\mathbf{R}_{gt}^{\top}) - 1}{2}), \qquad (7)$$

$$E_{\text{translation}} = ||\mathbf{t}_{pr} - \mathbf{t}_{gt}||_2. \qquad (8)$$

---

[3]It is worth noting that 3D Multi-view Registration [21] proposes both a method for pairwise registration and synchronizing multiple views at the same time. We only compare against their pairwise registration module.

We report the translation error in centimeters and the rotation errors in degrees. We also report the chamfer distance between the predicted and ground-truth alignments of the scene. For each metric, we report the mean and median errors as well as the accuracy at different thresholds.

Results. We first note that ICP approaches fail on this task. ICP assumes that the point clouds are prealigned and can be very effective at fine-tuning such alignment by minimizing a chamfer distance. However, our view pairs have a relatively large camera motion with the mean transformation between two views being 11.4 degrees and 19.4 cm. As a result, ICP struggles with the large transformations and partial overlap between the point cloud pairs. Similarly, FPFH also fails on this task as its output descriptors are not distinctive enough, resulting in many false correspondences which greatly deteriorates the registration performance.

On the other hand, learned approaches show a clear advantage in this domain as they are able to learn features that are well-tuned for the task and data domain. Our model is able to outperform FCGF despite FCGF being trained with ground-truth correspondences on an indoor scene dataset. This is true regardless of whether our model is trained using RGB-D or depth pairs. While we find that our model trained on 3D Match performs worse than FCGF, this is expected since 3DMatch is a much smaller dataset making it less suitable for a self-supervised approach.

Finally, our approach is competitive with approaches that use supervision for both feature learning and correspondence estimation [10, 21]. This comparison represents the difference between full supervision on a small dataset vs. self-supervision on a large dataset. Our competitive performance demonstrates the promise of self-supervision in this space and our model's ability to learn from a very simple learning signal: consistency between video frames.

	Rotation (deg)		Translation (cm)		Chamfer	
	Mean	Median	Mean	Median	Mean	Median
Random Visual	6.4	2.7	14.9	7.0	9.8	0.6
Random Geometric	21.3	13.0	46.5	28.5	26.0	8.6
BYOC (Visual)	2.7	0.9	6.4	2.6	3.3	0.1
BYOC-Geo	4.8	2.3	10.6	5.4	7.2	0.5
BYOC	3.8	1.7	8.7	4.3	5.6	0.3

Table 2. **Random Visual Features are surprisingly good for registration.** Random visual are better than geometric features for registration. This discrepancy persists after training.

What is the impact of the transformation estimator? While we observe that RANSAC improves the performance of FPFH and FCGF compared to the Weighted Procrustes, we see the opposite pattern with our approach. This is due to the fact that our model is trained specifically with a registration loss on filtered correspondence. As a result, Lowe's ratio becomes a very effective method of filtering our correspondences while being less effective for other approaches.

How good are random features? We find that random visual features can serve as a strong baseline for point cloud registration on ScanNet, as shown in Fig. 3 and Tab. 2. This is surprising since random visual features perform on-par with FCGF. This explains why our method is capable of achieving this performance without any supervision. We also find that after training, our visual features achieve the highest registration performance. Those results suggest that visual features are better descriptors for registration, but it is unclear if this a fundamental advantage or if the performance gap can be be resolved through better architectures or training schemes for geometric feature learning.

4.2. Correspondence Estimation

We now examine the quality of the correspondences estimated by our method. We evaluate our approach on the 3D Match geometric registration benchmark and follow the evaluation protocol proposed by Deng *et al*. [15] of evaluating the correspondence recall. Intuitively, feature-match recall measures the percentage of point cloud pairs that would be registered accurately using a RANSAC estimator by guaranteeing a minimum percentage of inliers.

Baselines. We compare our approach against three sets of baselines. The first set is hand-crafted features based on the local geometry around each point [44, 45, 49]. The second set is supervised approaches that use known pose to sample ground-truth correspondences and apply a metric learning loss to learn features for geometric registration. Finally, the third set is unsupervised approaches trained on reconstructed scenes. While those approaches do not directly use ground-truth pose during training, their training data (reconstructed scenes) is generated by aligning 50 depth maps

	Training Data		FMR	
	Dataset	Data Format	Recall	St. Dev.
SHOT [45]	-	-	0.238	0.109
USC [49]	-	-	0.400	0.125
FPFH [44]	-	-	0.481	0.150
FPFH [44] (corr)	-	-	0.462	0.198
3D Match [60]	3D Match	Depth + Pose	0.596	0.088
PPFNet [15]	3D Match	Depth + Pose	0.623	0.108
PerfectMatch [22]	3D Match	Depth + Pose	0.947	0.027
FCGF [12]	3D Match	Depth + Pose	0.952	0.066
FCGF [12] (corr)	3D Match	Depth + Pose	0.932	0.104
CGF [40]	SceneNN	Scenes	0.582	0.142
PPF-FoldNet [14]	3D Match	Scenes	0.718	0.105
3D PointCapsNet [64]	3D Match	Scenes	0.787	0.062
BYOC (no filtering)	ScanNet	RGB-D	0.662	0.225
BYOC	3D Match	RGB-D	0.690	0.172
BYOC	ScanNet	RGB-D	0.766	0.181
BYOC-Geo	ScanNet	Depth	0.786	0.195

Table 3. **Feature-Match Recall on 3D Match.** Our approach achieves better recall than hand-crafted and scene-supervised approaches while being competitive with supervised approaches.

into a single point cloud. Hence, while those approaches do not use pose supervision explicitly, pose information is *needed* to generate their data. We refer to those approaches as *scene-supervised*.

Evaluation Metrics. Given a set of correspondences $\mathcal{C}$, $FM(\mathcal{C})$ evaluates whether the percentage of inliers exceeds τ_2, where an inlier correspondence is defined as having a residual error less than τ_1 given the ground-truth transformation $\mathbf{T}^*$. Feature-match recall is the percentage of point cloud pairs that have a successful feature matching.

$$FM(\mathcal{C}) = \left[\frac{1}{|\mathcal{C}|} \sum_{(p,q) \in \mathcal{C}} \mathbb{1}\left(||\mathbf{x}_p - \mathbf{T}^* \mathbf{x}_q|| < \tau_1\right) \right] > \tau_2 \quad (9)$$

Similar to [12, 14, 15], we calculate feature-match recall over all view pairs using $\tau_1 = 10$ cm and $\tau_2 = 5\%$. Prior approaches often generate feature sets without any specified means of filtering them. As a result, they define the correspondence set as the set of all nearest neighbors. Unlike prior work, our approach outputs a small set of correspondences after ranking them using Lowe's ratio test.

Results. BYOC achieves high feature match recall, outperforming traditional and scene-supervised approaches, while being competitive with supervised approaches. This performance is achieved by only training on the raw RGB-D or depth scans without requiring any additional annotation or postprocessing of the data. This across dataset generalization is interesting since ScanNet and 3DMatch differ in two key ways. First, 3D Match point clouds are generated by integrating 50 depth frames. As a result, they are denser than single-frame ScanNet point clouds. Sec-
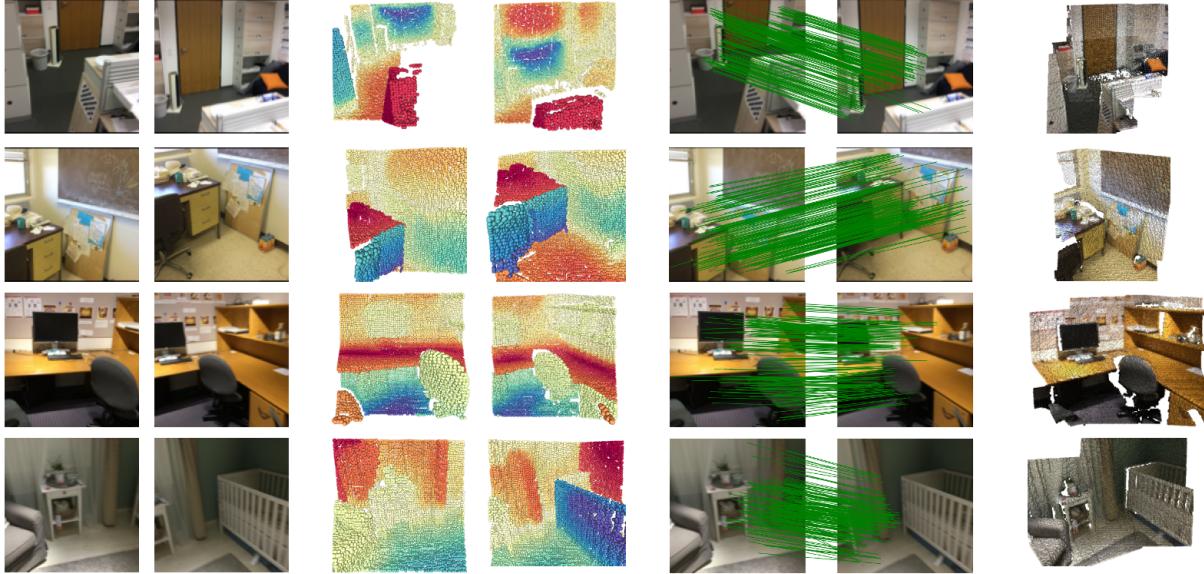
Figure 4. **BYOC's geometric features allow for accurate registration by mapping corresponding points to similar feature vectors.** Our approach learns informative geometric features of the scene. We visualize our features by mapping them to colors using t-SNE [53]. We find that the learned features appear to delineate objects such as chairs and floor edges. This results in the accurate registrations shown in the last column. Our approach takes uncolored point clouds as input; images and colored point clouds are presented to aid visualization.

ond, point cloud pairs in 3D Match have larger view point changes. Despite those differences, our model can still generalize from ScanNet to 3D Match. This can be attributed to both the voxelization performed by the geometric encoder and the augmentation which gives the model some degree of equivariance with respect to point cloud density and rotation.

We also observe that BYOC-Geo, which is only trained with geometric correspondence, generalizes better to 3D Match despite doing worse on ScanNet. One explanation for this discrepancy is that bootstrapping with visual correspondences biases the model towards representing features that are meaningful in both modalities. Such representations might be more dataset specific, hindering across-dataset generalization. This finding opens up the possibility of using datasets that only have depth video; *e.g.*, lidar.

While our best configuration performs on par with the best scene-supervised approach, they outperform us if we do not filter our correspondences. We observe that when we attempt to filter the correspondences for FPFH or FCGF, their performance deteriorates. This is consistent with some of the reported results by [14] where using a larger number of features improved their performance. Hence, it is unclear how correspondence filtering would affect the performance of self-supervised methods. Due to the lack of publicly-available implementations of those methods and the complexity of their approach, we were unable to run additional experiments to better understand the impact of the training data and correspondence filtering on the learning process.

5. Conclusion

We propose BYOC: a self-supervised approach to point cloud registration. Our key insight is that randomly initialized CNNs provide us with features that are good enough to bootstrap visual and geometric feature learning through point cloud registration. Our approach takes advantage of pseudo-correspondence labels that are obtained from the initially random encoders to train them using registration losses. We also show how non-contrastive learning can leverage the more accurate visual correspondences to learn better geometric features. At test time, we only use the geometric encoder to register point clouds without relying on any color or image information.

Our approach is both simple and fast: we rely on a fast sparse 3D convolutional encoder to extract features, use a ratio test to weigh and filter correspondences, and then align them using SVD. This deviates from current state-of-the-art approaches that use expensive prepossessing techniques [14, 15, 64], learn separate networks for correspondence estimation [10, 21, 40], and use RANSAC as the transformation estimation. Furthermore, we only use depth or RGB-D videos to train our model. This allows us to train on any dataset of such format, not only ones that can be accurately registered by traditional SfM pipelines.

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *CVPR*, 2019. 2

[2] KS Arun, TS Huang, and SD Blostein. Least square fitting of two 3-d point sets. In *TPAMI*, 1987. 2

[3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *CVPR*, 2020. 1, 2, 5

[4] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*. International Society for Optics and Photonics, 1992. 2

[5] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC: Differentiable RANSAC for Camera Localization. In *CVPR*, 2017. 2

[6] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *ICCV*, 2019. 2

[7] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 2

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 2, 3, 5

[9] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 1992. 2

[10] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, 2020. 2, 3, 4, 6, 8

[11] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019. 3, 5

[12] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, 2019. 1, 2, 3, 5, 6, 7

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 5

[14] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors. In *ECCV*, 2018. 2, 7, 8

[15] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *CVPR*, June 2018. 2, 7, 8

[16] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 3d local features for direct pairwise registration. In *CVPR*, 2019. 2

[17] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. *arXiv*, 2020. 2

[18] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2

[19] Mohamed El Banani, Luya Gao, and Justin Johnson. Unsupervisedr&r: Unsupervised point cloud registration via differentiable rendering. In *CVPR*, 2021. 2, 4

[20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[21] Zan Gojcic, Caifa Zhou, Jan D. Wegner, Leonidas J. Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *CVPR*, 2020. 2, 3, 4, 6, 8

[22] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *CVPR*, 2019. 2, 5, 6, 7

[23] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 4

[24] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, 2019. 2

[25] Benjamin Graham. Sparse 3D convolutional neural networks. In *BMVC*. Springer, 2015. 3

[26] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning. In *NeurIPS*, 2020. 2, 5

[27] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. Pointgmm: A neural gmm network for point clouds. In *CVPR*, 2020. 2

[28] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *CVPR*, 2020. 2

[29] Andrew E Johnson. *Spin-images: a representation for 3-D surface matching*. PhD thesis, Cargnegie Mellon University, 1997. 2

[30] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *TPAMI*, 1999. 2

[31] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 4

[32] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *ICCV*, 2017. 1, 2, 5

[33] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[34] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML - W*, 2013. 2

[35] Lei Li, Siyu Zhu, Hongbo Fu, Ping Tan, and Chiew-Lan Tai. End-to-end learning local multi-view descriptors for 3d point clouds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5

[36] Hugh Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981. 2

[37] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 4

[38] Hans P. Moravec. Rover visual obstacle avoidance. In *IJCAI*, 1981. 2

[39] François Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015. 2

[40] Rene Ranftl and Vladlen Koltun. Deep fundamental matrix

estimation. In *ECCV*, 2018. 2, 4, 7, 8

[41] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv*, 2020. 5

[42] Amir Rosenfeld and John K Tsotsos. Intriguing properties of randomly weighted networks: Generalizing while learning next to nothing. In *2019 16th Conference on Computer and Robot Vision (CRV)*, pages 9–16. IEEE, 2019. 2

[43] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001. 6

[44] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009. 2, 6, 7

[45] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 2014. 2, 7

[46] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[47] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1(1):1–5, 2017. 4

[48] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv*, 2019. 2

[49] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique shape context for 3d data description. In *Proceedings of the ACM Workshop on 3D Object Retrieval*, 3DOR '10. Association for Computing Machinery, 2010. 2, 7

[50] Philip HS Torr and David William Murray. The development and comparison of robust methods for estimating the fundamental matrix. *IJCV*, 1997. 2

[51] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018. 2

[52] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 4

[53] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008. 8

[54] Yue Wang and Justin Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *NeurIPS*, 2019. 2

[55] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, 2019. 2

[56] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pretraining for 3d point cloud understanding. In *ECCV*, 2020. 1, 2, 3, 5

[57] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *ECCV*, 2018. 1, 2, 5

[58] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *CVPR*, 2020. 2

[59] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jam-

pani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *ECCV*, 2020. 2

[60] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *CVPR*, 2017. 2, 5, 7

[61] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 1994. 2

[62] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 1995. 2

[63] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *arXiv preprint arXiv:2101.02691*, 2021. 2, 3, 5

[64] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3D Point Capsule Networks. In *CVPR*, 2019. 2, 7, 8

[65] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv*, 2018. 5