

Learning multiview 3D point cloud registration

Zan Gojcic^{*§}Caifa Zhou^{*§}Jan D. Wegner[§]Leonidas J. Guibas[†]Tolga Birdal[†][§]ETH Zurich[†]Stanford University

Abstract

We present a novel, end-to-end learnable, multiview 3D point cloud registration algorithm. Registration of multiple scans typically follows a two-stage pipeline: the initial pairwise alignment and the globally consistent refinement. The former is often ambiguous due to the low overlap of neighboring point clouds, symmetries and repetitive scene parts. Therefore, the latter global refinement aims at establishing the cyclic consistency across multiple scans and helps in resolving the ambiguous cases. In this paper we propose, to the best of our knowledge, the first end-to-end algorithm for joint learning of both parts of this two-stage problem. Experimental evaluation on well accepted benchmark datasets shows that our approach outperforms the state-of-the-art by a significant margin, while being end-to-end trainable and computationally less costly. Moreover, we present detailed analysis and an ablation study that validate the novel components of our approach. The source code and pretrained models are publicly available under https://github.com/zgojcic/3D_multiview_reg.

1. Introduction

Downstream tasks in 3D computer vision, such as semantic segmentation and object detection typically require a holistic representation of the scene. The capability of aligning and fusing individual point cloud fragments, which cover only small parts of the environment, into a globally consistent holistic representation is therefore essential and has several use cases in augmented reality and robotics. Pairwise registration of adjacent fragments is a well studied problem and traditional approaches based on geometric constraints [52, 69, 58] and hand-engineered feature descriptors [38, 27, 56, 61] have shown successful results to some extent. Nevertheless, in the recent years, research on local descriptors for pairwise registration of 3D point clouds is centered on deep learning approaches [70, 39, 21, 67, 19, 28] that succeed in capturing and encoding evidence hidden to hand-engineered descriptors. Furthermore, novel end-to-end methods for pairwise point cloud registration were

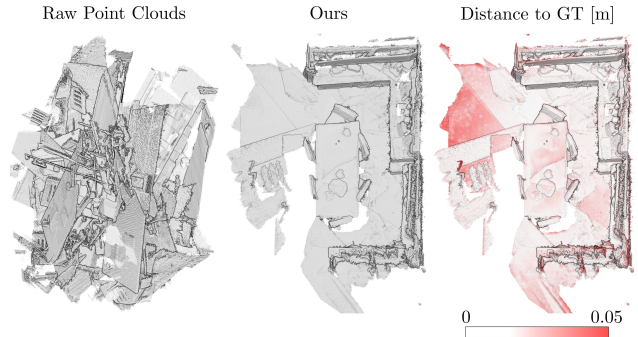


Figure 1. Result of our end-to-end reconstruction on the 60 scans of Kitchen scene from 3DMatch benchmark [70].

recently proposed [65, 43]. While demonstrating good performance for many tasks, pairwise registration of individual views of a scene has some conceptual drawbacks: (i) low overlap of adjacent point clouds can lead to inaccurate or wrong matches, (ii) point cloud registration has to rely on very local evidence, which can be harmful if 3D scene structure is scarce or repetitive, (iii) separate post-processing is required to combine all pair-wise matches into a global representation. Compared to the pairwise methods, globally consistent multiview alignment of unorganized point cloud fragments is yet to fully benefit from the recent advances achieved by the deep learning methods. State-of-the-art methods typically still rely on a good initialization of the pairwise maps, which they try to refine globally in a subsequent decoupled step [30, 63, 2, 3, 5, 4, 44, 11]. A general drawback of this hierarchical procedure is that global noise distribution over all nodes of the pose graph ends up being far from random, i.e. significant biases persist due to the highly correlated initial pairwise maps.

In this paper, we present, to the best of our knowledge, the first *end-to-end data driven multiview point cloud registration algorithm*. Our method takes a set of potentially overlapping point clouds as input and outputs a global/absolute transformation matrix per each of the input scans (*c.f.* Fig. 1). We depart from a traditional two-stage approach where the individual stages are detached from each other and directly learn to register all views of a scene in a globally consistent manner.

The main contributions of our work are:

- We formulate the traditional two-stage approach in an

^{*}First two authors contributed equally to this work.

end-to-end neural network, which in the forward pass solves two differentiable optimization problems: (i) the Procrustes problem for the estimation of the pairwise transformation parameters and (ii) the spectral relaxation of the transformation synchronization.

- We propose a confidence estimation block that uses a novel *overlap pooling* layer to predict the confidence in the estimated pairwise transformation parameters.
- We cast the multiview 3D point cloud registration problem as an iterative reweighted least squares (IRLS) problem and iteratively refine both the pairwise and absolute transformation estimates.

Resulting from the aforementioned contributions, the proposed multiview registration algorithm (i) is very efficient to compute, (ii) achieves more accurate scan alignments because the residuals are being fed back to the pairwise network in an iterative manner, (iii) outperforms current state-of-the-art on pairwise as well as multiview point cloud registration.

2. Related Work

Pairwise registration The traditional pairwise registration pipeline consists of two stages: the coarse alignment stage, which provides the initial estimate of the relative transformation parameters and the refinement stage that iteratively refines the transformation parameters by minimizing the 3D registration error under the assumption of rigid transformation.

The former is traditionally performed by using either handcrafted [56, 61, 60] or learned [70, 39, 21, 20, 67, 28, 16] 3D local features descriptors to establish the point-wise candidate correspondences in combination with a RANSAC-like robust estimator [26, 53, 41] or geometric hashing [24, 8, 33]. A parallel stream of works [1, 59, 45] relies on establishing correspondences using the 4-point congruent sets. In the refinement stage, the coarse transformation parameters are often fine-tuned with a variant of the iterative closest point (ICP) algorithm [6]. ICP-like algorithms [42, 66] perform optimization by alternatively hypothesizing the correspondence set and estimating the new set of transformation parameters. They are known to not be robust against outliers and to converge to a global optimum only when starting with a good prealignment [9]. ICP algorithms are often extended to use additional radiometric, temporal or odometry constraints [72]. Contemporary to our work, [65, 43] propose to integrate coarse and fine pairwise registration stages into an end-to-end learnable algorithm. Using a deep network, [31] formulates the object tracking as a relative motion estimation of two point sets.

Multiview registration Multiview, global point cloud registration methods aim at resolving hard or ambiguous cases

that arise in pairwise methods by incorporating cues from multiple views. The first family of methods employ a multiview ICP-like scheme to optimize for camera poses as well as 3D point correspondences [37, 25, 46, 9]. A majority of these suffer from increased complexity of correspondence estimation. To alleviate this, some approaches only optimize for motion and use the scans to evaluate the registration error [72, 58, 7]. Taking a step further, other modern methods make use of the global cycle-consistency and optimize only over the poses starting from an initial set of pairwise maps. This efficient approach is known as *synchronization* [10, 63, 2, 58, 3, 5, 44, 72, 7, 36]. Global structure-from-motion [17, 73] aims to synchronize the observed relative motions by decomposing rotation, translation and scale components. [23] proposes a global point cloud registration approach using two networks, one for pose estimation and another modelling the scene structure by estimating the occupancy status of global coordinates.

Probably the most similar work to ours is [36], where the authors aim to adapt the edge weights for the transformation synchronization layer by learning a data driven weighting function. A major conceptual difference to our approach is that relative transformation parameters are estimated using FPFH [56] in combination with FGR [72] and thus, unlike ours, are not learned. Furthermore, in each iteration [36] has to convert the point clouds to depth images as the weighting function is approximated by a 2D CNN. On the other hand our whole approach operates directly on point clouds, is fully differentiable and therefore facilitates learning a global, multiview point cloud registration in an end-to-end manner.

3. End-to-End Multiview 3D Registration

In this section we derive the proposed multiview 3D registration algorithm as a composition of functions depending upon the data. The network architectures used to approximate these functions are then explained in detail in Sec 4. We begin with a new algorithm for learned pairwise point cloud registration, which uses two point clouds as input and outputs estimated transformation parameters (Sec. 3.1). This method is extended to multiple point clouds by using a transformation synchronization layer amenable to back-propagation (Sec. 3.2). The input graph to this synchronization layer encodes, along with the relative transformation parameters, the confidence in these pairwise maps, which is also estimated using a novel neural network, as edge information. Finally, we propose an IRLS scheme (Sec. 3.3) to refine the global registration of all point clouds by updating the edge weights as well as the pairwise poses.

Consider a set of potentially overlapping point clouds $S = \{\mathbf{S}_i \in \mathbb{R}^{N \times 3}, 1 \leq i \leq N_S\}$ capturing a 3D scene from different viewpoints (i.e. poses). The task of *multiview registration* is to recover the rigid, absolute poses

$\{\mathbf{M}_i^* \in SE(3)\}_i$ given the scan collection, where

$$SE(3) = \left\{ \mathbf{M} \in \mathbb{R}^{4 \times 4} : \mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \right\}, \quad (1)$$

$\mathbf{R}_i \in SO(3)$ and $\mathbf{t}_i \in \mathbb{R}^3$. \mathcal{S} can be augmented by connectivity information resulting in a finite graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, where each vertex represents a single point set and the edges $(i, j) \in \mathcal{E}$ encode the information about the relative rotation \mathbf{R}_{ij} and translation \mathbf{t}_{ij} between the vertices. These relative transformation parameters satisfy $\mathbf{R}_{ij} = \mathbf{R}_{ji}^T$ and $\mathbf{t}_{ij} = -\mathbf{R}_{ij}^T \mathbf{t}_{ji}$ as well as the *compatibility constraint* [4]

$$\mathbf{R}_{ij} \approx \mathbf{R}_i \mathbf{R}_j^T \quad \mathbf{t}_{ij} \approx -\mathbf{R}_i \mathbf{R}_j^T \mathbf{t}_j + \mathbf{t}_i \quad (2)$$

In current state-of-the-art [72, 36, 7] edges \mathcal{E} of \mathcal{G} are initialized with (noisy) relative transformation parameters $\{\mathbf{M}_{ij}\}$, obtained by an independent, auxiliary pairwise registration algorithm. Global scene consistency is enforced via a subsequent synchronization algorithm. In contrast, we propose a joint approach where pairwise registration and transformation synchronization are tightly coupled as one fully differentiable component, which leads to an end-to-end learnable, global registration pipeline.

3.1. Pairwise registration of point clouds

In the following, we introduce a differentiable, pairwise registration algorithm that can easily be incorporated into an end-to-end multiview 3D registration algorithm. Let $\{\mathbf{P}, \mathbf{Q}\} := \{\mathbf{S}_i, \mathbf{S}_j | i \neq j\} \subset \mathcal{S}$ denote a pair of point clouds where $(\mathbf{P})_l =: \mathbf{p}_l \in \mathbb{R}^3$ and $(\mathbf{Q})_l =: \mathbf{q}_l \in \mathbb{R}^3$ represent the coordinate vectors of individual points in point clouds $\mathbf{P} \in \mathbb{R}^{N_P \times 3}$ and $\mathbf{Q} \in \mathbb{R}^{N_Q \times 3}$, respectively. The goal of pairwise registration is to retrieve optimal $\hat{\mathbf{R}}_{ij}$ and $\hat{\mathbf{t}}_{ij}$.

$$\hat{\mathbf{R}}_{ij}, \hat{\mathbf{t}}_{ij} = \arg \min_{\mathbf{R}_{ij}, \mathbf{t}_{ij}} \sum_{l=1}^{N_P} \|\mathbf{R}_{ij} \mathbf{p}_l + \mathbf{t}_{ij} - \phi(\mathbf{p}_l, \mathbf{Q})\|^2 \quad (3)$$

where $\phi(\mathbf{p}, \mathbf{Q})$ is a *correspondence function* that maps the points \mathbf{p} to their corresponding points in point cloud \mathbf{Q} . The formulation of Eq. 3 facilitates a differentiable closed-form solution, which is—subject to the noise distribution—close to the ground truth solution [57]. However, least square solutions are not robust and thus Eq. 3 will yield wrong transformation parameters in case of high outlier ratio. In practice, the mapping $\phi(\mathbf{p}, \mathbf{Q})$ is far from ideal and erroneous correspondences typically dominate. To circumvent that, Eq. 3 can be robustified against outliers by introducing a heteroscedastic weighting matrix [62, 57]:

$$\hat{\mathbf{R}}_{ij}, \hat{\mathbf{t}}_{ij} = \arg \min_{\mathbf{R}_{ij}, \mathbf{t}_{ij}} \sum_{l=1}^{N_P} w_l \|\mathbf{R}_{ij} \mathbf{p}_l + \mathbf{t}_{ij} - \phi(\mathbf{p}_l, \mathbf{Q})\|^2 \quad (4)$$

where $w_l := (\mathbf{w})_l$ is the weight of the putative correspondence $\gamma_l \in \mathbb{R}^6 = \{\mathbf{p}_l, \phi(\mathbf{p}_l, \mathbf{Q})\}$ computed by some weighting function $\mathbf{w} = \psi_{\text{init}}(\mathbf{\Gamma})$, where $\mathbf{\Gamma} := \{\gamma_l\} := \{\mathbf{P}, \{\phi(\mathbf{p}_l, \mathbf{Q})\}_l\}$ and $\psi_{\text{init}} : \mathbb{R}^{N_P \times 6} \mapsto \mathbb{R}^{N_P}$. Assuming that w_l is close to one when the putative correspondence is an inlier and close to zero otherwise, Eq. 4 will yield the correct transformation parameters while retaining a differentiable closed-form solution [57]. Hereinafter we denote this closed-form solution as weighted least squares transformation *WLS trans.* and for the sake of completeness, its derivation is provided in the supp. material.

3.2. Differentiable transformation synchronization

Returning to the task of multiview registration, we again consider the initial set of point clouds \mathcal{S} . If no prior connectivity information is given, graph \mathcal{G} can be initialized by forming $\binom{N_S}{2}$ point cloud pairs and estimating their relative transformation parameters as described in Sec. 3.1. The global transformation parameters can be estimated either jointly (*transformation synchronization*) [30, 5, 4, 11] or by dividing the problem into *rotation synchronization* [2, 3] and *translation synchronization* [35]. Herein, we opt for the latter approach, which under the spectral relation admits a differentiable closed-form solution [2, 3, 35].

Rotation synchronization The goal of rotation synchronization is to retrieve global rotation matrices $\{\mathbf{R}_i^*\}$ by solving the following minimization problem based on their observed ratios $\{\hat{\mathbf{R}}_{ij}\}$

$$\mathbf{R}_i^* = \arg \min_{\mathbf{R}_i \in SO(3)} \sum_{(i,j) \in \mathcal{E}} c_{ij} \|\hat{\mathbf{R}}_{ij} - \mathbf{R}_i \mathbf{R}_j^T\|_F^2 \quad (5)$$

where the weights $c_{ij} := \zeta_{\text{init}}(\mathbf{\Gamma})$ represent the confidence in the relative transformation parameters $\hat{\mathbf{M}}_{ij}$. Under the spectral relaxation Eq. 5 admits a closed-form solution, which is provided in the supp. material [2, 3].

Translation synchronization Similarly, the goal of translation synchronization is to retrieve global translation vectors $\{\mathbf{t}_i^*\}$ that minimize the following least squares problem

$$\mathbf{t}_i^* = \arg \min_{\mathbf{t}_i} \sum_{(i,j) \in \mathcal{E}} c_{ij} \|\hat{\mathbf{R}}_{ij} \mathbf{t}_i + \hat{\mathbf{t}}_{ij} - \mathbf{t}_j\|^2 \quad (6)$$

The differentiable closed-form solution to Eq. 6 is again provided in the supp. material.

3.3. Iterative refinement of the registration

The above formulation (Sec. 3.1 and 3.2) facilitates an implementation in an iterative scheme, which in turn can be viewed as an IRLS algorithm. We can start each subsequent iteration $(k+1)$ by pre-aligning the point cloud pairs using the synchronized estimate of the relative transforma-

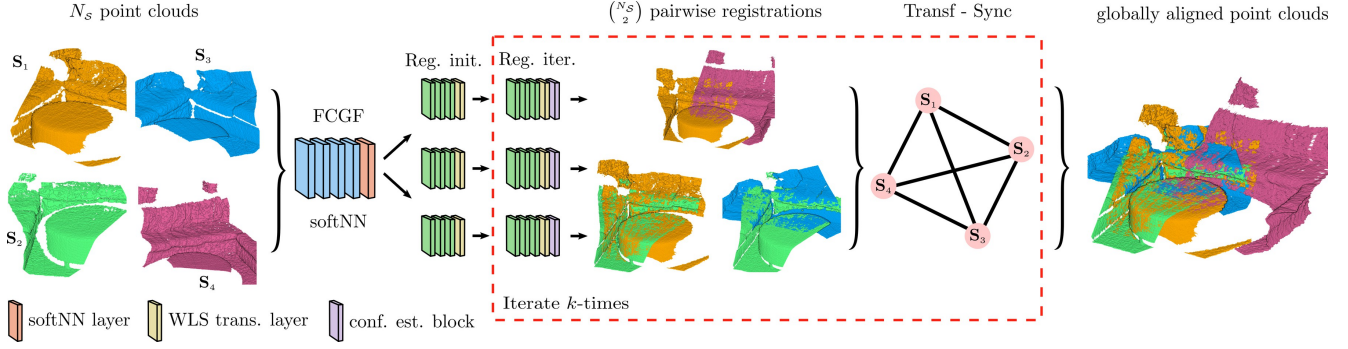


Figure 2. Proposed pipeline for end-to-end multiview 3D point cloud registration. For each of the input point clouds S_i we extract FCGF [16] features that are fed to the softNN layer to compute the stochastic correspondences for $\binom{N_s}{2}$ pairs. These correspondences are used as input to the initial registration block (i.e. *Reg. init.*) that outputs the per-correspondence weights, initial transformation parameters, and per-point residuals. Along with the correspondences, the initial weights and residuals are then input to the registration refinement block (i.e. *Reg. iter.*), whose outputs are used to build the graph. After each iteration of the *Transf-Sync* layer the estimated transformation parameters are used to pre-align the correspondences that are concatenated with the weights from the previous iteration and the residuals and feed anew to *Reg. iter.* block. We iterate over the *Reg. iter.* and *Transf-Sync* layer for four times.

tion parameters $\mathbf{M}_{ij}^{*(k)} = \mathbf{M}_i^{*(k)} \mathbf{M}_j^{*(k)-1}$ from iteration (k) such that $\mathbf{Q}^{(k+1)} := \mathbf{M}_{ij}^{*(k)} \otimes \mathbf{Q}$, where \otimes denotes applying the transformation $\mathbf{M}_{ij}^{*(k)}$ to point cloud \mathbf{Q} . Additionally, weights $\mathbf{w}^{(k)}$ and residuals $\mathbf{r}^{(k)}$ of the previous iteration can be used as a side information in the correspondence weighting function. Therefore, $\psi_{\text{init}}(\cdot)$ is extended to

$$\mathbf{w}^{(k+1)} := \psi_{\text{iter}}(\mathbf{\Gamma}^{(k+1)}, \mathbf{w}^{(k)}, \mathbf{r}^{(k)}), \quad (7)$$

where $\mathbf{\Gamma}^{(k+1)} := \{\gamma_l^{(k+1)}\} := \{\mathbf{P}, \{\phi(\mathbf{p}_l, \mathbf{Q}^{(k+1)})\}_l\}$. Analogously, the difference between the input $\hat{\mathbf{M}}_{ij}^{(k)}$ and the synchronized $\mathbf{M}_{ij}^{*(k)}$ transformation parameters of the (k) -th iteration can be used as an additional cue for estimating the confidence $c_{ij}^{(k+1)}$. Thus, $\zeta_{\text{init}}(\cdot)$ can be extended to

$$c_{ij}^{(k+1)} := \zeta_{\text{iter}}(\mathbf{\Gamma}^{(k+1)}, \hat{\mathbf{M}}_{ij}^{(k)}, \mathbf{M}_{ij}^{*(k)}). \quad (8)$$

4. Network Architecture

We implement our proposed *multiview registration* algorithm as a deep neural network (Fig. 2). In this section, we first describe the architectures used to approximate $\phi(\cdot)$, $\psi_{\text{init}}(\cdot)$, $\psi_{\text{iter}}(\cdot)$, $\zeta_{\text{init}}(\cdot)$ and $\zeta_{\text{iter}}(\cdot)$, before integrating them into one fully differentiable, end-to-end trainable algorithm.

Learned correspondence function Our approximation of the correspondence function $\phi(\cdot)$ extends a recently proposed fully convolutional 3D feature descriptor FCGF [16] with a soft assignment layer. FCGF operates on sparse tensors [15] and computes 32 dimensional descriptors for each point of the sparse point cloud in a single pass. Note that the function $\phi(\cdot)$ could be approximated with any of the recently proposed learned feature descriptors [39, 20, 21, 28],

but we choose FCGF due to its high accuracy and low computational complexity.

Let \mathbf{F}_P and \mathbf{F}_Q denote the FCGF embeddings of point clouds P and Q obtained using the same network weights, respectively. Pointwise correspondences $\{\phi(\cdot)\}$ can then be established by a nearest neighbor (NN) search in this high dimensional feature space. However, the selection rule of such hard assignments is not differentiable. We therefore form the NN-selection rule in a probabilistic manner by computing a probability vector \mathbf{s} of the categorical distribution [50]. The stochastic correspondence of the point \mathbf{p} in the point cloud Q is then defined as

$$\phi(\mathbf{p}, \mathbf{Q}) := \mathbf{s}^T \mathbf{Q}, \quad (\mathbf{s})_l := \frac{\exp(-d_l/t)}{\sum_{l=1}^{N_Q} \exp(-d_l/t)} \quad (9)$$

where $d_l := \|\mathbf{f}_p - (\mathbf{F}_Q)_l\|_2$, \mathbf{f}_p is the FCGF embedding of the point \mathbf{p} and t denotes the temperature parameter. In the limit $t \rightarrow 0$ the $\phi(\mathbf{p}, \mathbf{Q})$ converges to the deterministic NN-search [50].

We follow [16] and supervise the learning of $\phi(\cdot)$ with a correspondence loss \mathcal{L}_c , which is defined as the hardest contrastive loss and operates on the FCGF embeddings

$$\mathcal{L}_c = \frac{1}{N_{\text{FCGF}}} \sum_{(i,j) \in \mathcal{P}} \left\{ [d(\mathbf{f}_i, \mathbf{f}_j) - m_p]_+^2 / |\mathcal{P}| + 0.5 [m_n - \min_{k \in \mathcal{N}} d(\mathbf{f}_i, \mathbf{f}_k)]_+^2 / |\mathcal{N}_i| + 0.5 [m_n - \min_{k \in \mathcal{N}} d(\mathbf{f}_j, \mathbf{f}_k)]_+^2 / |\mathcal{N}_j| \right\}$$

where \mathcal{P} is a set of all the positive pairs in a FCGF mini batch N_{FCGF} and \mathcal{N} is a random subset of all features that is used for the hardest negative mining. $m_p = 0.1$ and $m_n = 1.4$ are the margins for positive and negative pairs

respectively. The detailed network architecture of $\phi(\cdot)$ as well as the training configuration and parameters are available in the supp. material.

Deep pairwise registration Despite the good performance of the FCGF descriptor, several putative correspondences $\Gamma' \subset \Gamma$ will be false. Furthermore, the distribution of inliers and outliers does not resemble noise but rather shows regularity [54]. We thus aim to learn this regularity from the data using a deep neural network. Recently, several networks representing a complex weighting function for filtering of 2D [47, 54, 71] or 3D [29] feature correspondences have been proposed.

Herein, we propose extending the 3D outlier filtering network [29] that is based on [47] with the order-aware blocks proposed in [71]. Specifically, we create a pairwise registration block $f_\theta : \mathbb{R}^{N_P \times 6} \mapsto \mathbb{R}^{N_P}$ that takes the coordinates of the putative correspondences Γ as input and outputs weights $\mathbf{w} := \psi_{\text{init}}(\Gamma) := \tanh(\text{ReLU}(f_\theta(\Gamma)))$ that are fed, along with Γ , into the closed form solution of Eq. 4 to obtain $\hat{\mathbf{R}}_{ij}$ and $\hat{\mathbf{t}}_{ij}$. Motivated by the results in [54, 71] we add another registration block $\psi_{\text{iter}}(\cdot)$ to our network and append the weights \mathbf{w} and the pointwise residuals \mathbf{r} to the original input s.t. $\mathbf{w}^{(k)} := \psi_{\text{iter}}(\text{cat}([\Gamma^{(k)}, \mathbf{w}^{(k-1)}, \mathbf{r}^{(k-1)}]))$ (see Sec. 3.3). The weights $\mathbf{w}^{(k)}$ are then, again fed together with the initial correspondences Γ to the closed form solution of Eq. 4 to obtain the refined pairwise transformation parameters. In order to ensure permutation-invariance of $f_\theta(\cdot)$ a PointNet-like [51] architecture that operates on individual correspondences is used in both registration blocks. As each branch only operates on individual correspondences, the local 3D context information is gathered in the intermediate layers using symmetric context normalization [68] and order-aware filtering layers [71]. The detailed architecture of the registration block is available in the supp. material. Training of the registration network is supervised using the registration loss \mathcal{L}_{reg} defined for a batch with N_{reg} examples as

$$\mathcal{L}_{\text{reg}} = \alpha_{\text{reg}} L_{\text{class}} + \beta_{\text{reg}} L_{\text{trans}} \quad (10)$$

loss, where $\mathcal{L}_{\text{class}}$ denotes the binary cross entropy loss and

$$\mathcal{L}_{\text{trans}} = \frac{1}{N_{\text{reg}}} \sum_{(i,j)} \frac{1}{N_P} \sum_{l=1}^{N_P} \|\hat{\mathbf{M}}_{ij} \otimes \mathbf{p}_l - \mathbf{M}_{ij}^{\text{GT}} \otimes \mathbf{p}_l\|_2 \quad (11)$$

is used to penalize the deviation from the ground truth transformation parameters $\mathbf{M}_{ij}^{\text{GT}}$. α_{reg} and β_{reg} are used to control the contribution of the individual loss functions.

Confidence estimation block Along with the estimated relative transformation parameters $\hat{\mathbf{M}}_{ij}$, the edges of the graph \mathcal{G} encode the confidence c_{ij} in those estimates. Confidence encoded in each edge of the graph consist of (i) the

local confidence c_{ij}^{local} of the pairwise transformation estimation and (ii) the global confidence c_{ij}^{global} derived from the transformation synchronization. We formulate the estimation of c_{ij}^{local} as a classification task and argue that some of the required information is encompassed in the features of the second-to-last layer of the registration block. Let $\mathbf{X}_{ij}^{\text{conf}} = f_\theta^{(-2)}(\cdot)$ denote the output of the second-to-last layer of the registration block, we propose an *overlap pooling layer* f_{overlap} that extracts a global feature $\mathbf{x}_{ij}^{\text{conf}}$ by performing the weighted average pooling as

$$\mathbf{x}_{ij}^{\text{conf}} = \mathbf{w}_{ij}^T \mathbf{X}_{ij}^{\text{conf}}. \quad (12)$$

The obtained global feature is concatenated with the ratio of inliers δ_{ij} (i.e., the number of correspondences whose weights are higher than a given threshold) and fed to the confidence estimation network with three fully connected layers (129 – 64 – 32 – 1), followed by a ReLU activation function. The local confidence can thus be expressed as

$$c_{ij}^{\text{local}} := \zeta_{\text{init}}(\Gamma) := \text{MLP}(\text{cat}([\mathbf{x}_{ij}^{\text{conf}}, \delta_{ij}])) \quad (13)$$

The training of the confidence estimation block is supervised with the confidence loss function $\mathcal{L}_{\text{conf}} = \frac{1}{N} \sum_{(i,j)} \text{BCE}(c_{ij}^{\text{local}}, c_{ij}^{\text{GT}})$ (N denotes the number of cloud pairs), where BCE refers to the binary cross entropy and the ground truth confidence c_{ij}^{GT} labels are computed on the fly by thresholding the angular error $\tau_a = \arccos(\frac{\text{Tr}(\hat{\mathbf{R}}_{ij}^T \mathbf{R}_{ij}^{\text{GT}}) - 1}{2})$.

The $\zeta_{\text{init}}(\cdot)$ incorporates the *local* confidence in the relative transformation parameters. On the other hand, the output of the transformation synchronization layer provides the information how the input relative transformations agree globally with the other edges. In fact, traditional synchronization algorithms [13, 4, 35] only use this *global* information to perform the reweighting of the edges in the iterative solutions, because they do not have access to the *local* confidence information. *Global* confidence in the relative transformation parameters c_{ij}^{global} can be expressed with the Cauchy weighting function [34, 4]

$$c_{ij}^{\text{global}} = 1/(1 + r_{ij}^*/b) \quad (14)$$

where $r_{ij}^* = \|\hat{\mathbf{M}}_{ij} - \mathbf{M}_i^* \mathbf{M}_j^{*T}\|_F$ and following [34, 4] $b = 1.482 \gamma \text{ med}(|\mathbf{r}^* - \text{med}(\mathbf{r}^*)|)$ with $\text{med}(\cdot)$ denoting the median operator and \mathbf{r}^* the vectorization of residuals \mathbf{r}_{ij}^* . Since *local* and *global* confidence provide complementary information about the relative transformation parameters, we combine them into a joined confidence c_{ij} using their harmonic mean:

$$c_{ij} := \zeta_{\text{iter}}(c_{ij}^{\text{local}}, c_{ij}^{\text{global}}) := \frac{(1 + \beta^2) c_{ij}^{\text{global}} \cdot c_{ij}^{\text{local}}}{\beta^2 c_{ij}^{\text{global}} + c_{ij}^{\text{local}}} \quad (15)$$

	<i>3DMatch</i> [70]	<i>CGF</i> [39]	<i>PPFNet</i> [21]	<i>3DR</i> [22]	<i>3DSN</i> [28]	<i>FCGF</i> [16]	<i>Ours</i>	
							<i>1-iter</i>	<i>4-iter</i>
Kitchen	0.85	0.72	0.90	0.80	0.96	0.95	0.96	0.98
Home 1	0.78	0.69	0.58	0.81	0.88	0.91	0.92	0.93
Home 2	0.61	0.46	0.57	0.70	0.79	0.72	0.70	0.73
Hotel 1	0.79	0.55	0.75	0.73	0.95	0.93	0.95	0.97
Hotel 2	0.59	0.49	0.68	0.67	0.83	0.88	0.90	0.90
Hotel 3	0.58	0.65	0.88	0.94	0.92	0.81	0.89	0.89
Study	0.63	0.48	0.68	0.70	0.84	0.86	0.86	0.92
MIT Lab	0.51	0.42	0.62	0.62	0.76	0.82	0.78	0.78
Average	0.67	0.56	0.71	0.75	0.86	0.86	0.87	0.89

Table 1. Registration recall on 3DMatch data set. *1-iter* and *4-iter* denote the result of the pairwise registration network and input to the 4th Transf-Sync layer, respectively. Best results, except for *4-iter* that is informed by the global information, are shown in bold.

where the β balances the contribution of the local and global confidence estimates and is learned during training.

End-to-end multiview 3D registration The individual parts of the network are connected into an end-to-end multiview 3D registration algorithm as shown in Fig. 2². We pre-train the individual sub-networks (training details available in the supp. material) before fine-tuning the whole model in an end-to-end manner on the 3DMatch data set [70] using the official train/test data split. In fine-tuning we use $N_{FCGF} = 4$ to extract the FCGF features and randomly sample feature vectors of 2048 points per fragment. These features are used in the soft assignment (softNN) to form the putative correspondences of $\binom{N_s}{2}$ point clouds pairs³, which are fed to the pairwise registration network. The output of the pairwise registration is used to build the graph, which is input to the transformation synchronization layer. The iterative refinement of the transformation parameters is performed four times. We supervise the fine tuning using the joint multiview registration loss

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_{reg} + \mathcal{L}_{conf} + \mathcal{L}_{sync} \quad (16)$$

where the transformation synchronization \mathcal{L}_{sync} loss reads

$$\mathcal{L}_{sync} = \frac{1}{N} \sum_{(i,j)} (\|\mathbf{R}_{ij}^* - \mathbf{R}_{ij}^{GT}\|_F + \|\mathbf{t}_{ij}^* - \mathbf{t}_{ij}^{GT}\|_2). \quad (17)$$

We fine-tune the whole network for 2400 iterations using Adam optimizer [40] with a learning rate of 5×10^{-6} .

5. Experiments

We conduct the evaluation of our approach on the publicly available benchmark datasets *3DMatch* [70], *Redwood* [14] and *ScanNet* [18]. First, we evaluate the performance, efficiency, and the generalization capacity of the proposed pairwise registration algorithm on *3DMatch* and

²The network is implemented in Pytorch [48]. A pseudo-code of the proposed approach is provided in the supp. material.

³We assume a fully connected graph during training but are able to consider the connectivity information, if provided.

	Per fragment pair		Whole scene
	NN search [s]	Model estimation [s]	Total time [s]
RANSAC	0.38	0.23	1106.3
Ours (softNN)	0.10	0.01	80.3

Table 2. Average run-time for estimating the pairwise transformation parameters of one fragment pair on *3DMatch* dataset. Note, the GPU implementation of the soft assignments is faster than the CPU based kd-tree NN search.

Redwood dataset respectively (Sec. 5.1). We then evaluate the whole pipeline on the global registration of the point cloud fragments generated from RGB-D images, which are part of the *ScanNet* dataset [18].

5.1. Pairwise registration performance

We begin by evaluating the pairwise registration part of our algorithm on a traditional geometric registration task. We compare the results of our method to the state-of-the-art data-driven feature descriptors *3DMatch* [70], *CGF* [39], *PPFNet* [21], *3DSmoothNet* (3DS) [28], and *FCGF* [16], which is also used as part of our algorithm, as well as to a recent network based registration algorithm *3DR* [22]. Following the evaluation procedure of *3DMatch* [70] we complement all the descriptor based methods with the RANSAC-based transformation parameter estimation. For our approach we report the results after the pairwise registration network (1-iter in Tab. 1) as well as the the output of the $\psi_{iter}(\cdot)$ in the 4th iteration (4-iter in Tab. 1). The latter is already informed with the global information and serves primarily as verification that with the iterations our input to the *Transf-Sync* layer improves. Consistent with the *3DMatch* evaluation procedure, we report the average recall per scene as well as for the whole dataset in Tab. 1.

The registration results show that our approach reaches the highest recall among all the evaluated methods. More importantly, it indicates that using the same features (FCGF), our method can outperform RANSAC-based estimation of the transformation parameters, while having a much lower time complexity (Tab. 2). The comparison of the results of 1-iter and 4-iter also confirms the intuition that feeding the residuals and weights of the previous estimation back to the pairwise registration block helps refining the estimated pairwise transformation parameters.

Generalization to other domains In order to test if our pairwise registration model can generalize to new datasets and unseen domains, we perform a generalization evaluation on a synthetic indoor dataset *Redwood indoor* [14]. We follow the evaluation protocol of [14] and report the average registration recall and precision across all four scenes. We compare our approach to the recent data driven approaches *3DMatch* [70], *CGF* [39]+*FGR* [72] or *CZK* [14], *RelativeNet* (RN) [22], *3DR* [22] and traditional methods *CZK* [14] and *Latent RANSAC* (LR) [41]. Fig. 3 shows

Methods		Rotation Error						Translation Error (m)					
		3°	5°	10°	30°	45°	Mean/Med.	0.05	0.1	0.25	0.5	0.75	Mean/Med.
Pairwise (All)	<i>FGR</i> [72]	9.9	16.8	23.5	31.9	38.4	76.3°/-	5.5	13.3	22.0	29.0	36.3	1.67/-
	<i>Ours</i> (1 st iter.)	32.6	37.2	41.0	46.5	49.4	65.9°/48.8°	25.1	34.1	40.0	43.4	46.8	1.37/0.94
Edge Pruning (All)	<i>Ours</i> (4 th iter.)	34.3	38.7	42.2	48.2	51.9	62.3°/37.0°	26.7	35.7	41.8	45.5	49.4	1.26/0.78
	<i>Ours</i> (After Sync.)	40.7	45.7	50.8	56.2	58.4	52.2°/9.0°	29.3	42.1	50.9	54.7	58.3	0.96/0.20
FGR (Good)	<i>FastGR</i> [72]	12.4	21.4	29.5	38.6	45.1	68.8°/-	7.7	17.6	28.2	36.2	43.4	1.43/-
	<i>GeoReg</i> (FGR) [14]	0.2	0.6	2.8	16.4	27.1	87.2°/-	0.1	0.7	4.8	16.4	28.4	1.80/-
	<i>EIGSE3</i> (FGR) [4]	1.5	4.3	12.1	34.5	47.7	68.1°/-	1.2	4.1	14.7	32.6	46.0	1.29/-
	<i>RotAvg</i> (FGR) [12]	6.0	10.4	17.3	36.1	46.1	64.4°/-	3.7	9.2	19.5	34.0	45.6	1.26/-
	<i>L2Sync</i> (FGR) [36]	34.4	41.1	49.0	58.9	62.3	42.9°/-	2.0	7.3	22.3	36.9	48.1	1.16/-
	<i>Ours</i> (After Sync.)	65.8	72.8	77.6	81.9	83.2	20.3°/1.6°	48.4	67.2	76.5	79.7	82.0	0.42/0.05
Ours (Good)	<i>EIGSE3</i> [4]	63.3	70.2	75.6	80.5	81.6	23.0°/1.7°	42.2	58.5	69.8	76.9	79.7	0.45/0.06
	<i>Ours</i> (1 st iter.)	57.7	65.5	71.3	76.5	78.1	28.3°/1.9°	44.8	60.3	69.6	73.1	75.5	0.57/0.06
	<i>Ours</i> (4 th iter.)	60.6	68.3	73.7	78.9	81.0	24.2°/1.8°	47.1	63.3	72.2	76.2	78.7	0.50/0.05
	<i>Ours</i> (After Sync.)	65.8	72.8	77.6	81.9	83.2	20.3°/1.6°	48.4	67.2	76.5	79.7	82.0	0.42/0.05

Table 3. Multiview registration evaluation on *ScanNet* [18] dataset. We report the ECDF values for rotation and translation errors. Best results are shown in bold.

that our approach can achieve ≈ 4 percentage points higher recall than state-of-the-art without being trained on synthetic data and thus confirming the good generalization capacity of our approach. Note that while the average precision across the scenes is low for all the methods, several works [14, 39, 22] show that the precision can easily be increased using pruning without almost any loss in the recall.

5.2. Multiview registration performance

We finally evaluate the performance of our complete method on the task of multiview registration using the *ScanNet* [18] dataset. *ScanNet* is a large RGBD dataset of indoor scenes. It provides the reconstructions, ground truth camera poses and semantic segmentations for 1513 scenes. To ensure a fair comparison, we follow [36] and use the same 32 randomly sampled scenes for evaluation. For each scene we randomly sample 30 RGBD images that are 20 frames apart and convert them to point clouds. The temporal sequence of the frames is discarded. In combination with the large temporal gap between the frames, this makes the test setting extremely challenging. Different to [36], we do not train our network on *ScanNet*, but rather perform direct generalization of the network trained on the 3DMatch dataset.

Evaluation protocol We use the standard evaluation protocol [13, 36] and report the empirical cumulative distribution function (ECDF) for the angular a_e and translation t_e deviations defined as

$$a_e = \arccos\left(\frac{\text{Tr}(\mathbf{R}_{ij}^{*T} \mathbf{R}_{ij}^{\text{GT}}) - 1}{2}\right) \quad t_e = \|\mathbf{t}_{ij}^{\text{GT}} - \mathbf{t}_{ij}^*\|_2 \quad (18)$$

The ground truth rotations \mathbf{R}^{GT} and translations \mathbf{t}^{GT} are provided by the authors of *ScanNet* [18]. In Tab. 3 we report the results for three different scenarios. "FGR (Good)" and "Ours (Good)" denote the scenarios in which we follow [36] and use the computed pairwise registrations to prune the edges before the transformation synchronization

if the median point distance in the overlapping⁴ region after the transformation is larger than 0.1m (FGR) or 0.05m (ours). The EIGSE3 in "Ours (Good)" is initialized using our pairwise estimates. On the other hand, "all" denotes the scenario in which all $\binom{N_S}{2}$ pairs are used to build the graph. In all scenarios we prune the edges of the graph if the confidence estimation in the relative transformation parameters of that edge c_{ij}^{local} drops below $\tau_p = 0.85$. This threshold was determined on 3DMatch dataset and its effect on the performance of our approach is analyzed in detail in the supp. material. If during the iterations the pruning of the edges yields a disconnected graph we simply report the last valid values for each node before the graph becomes disconnected. A more sophisticated handling of the edge pruning and disconnected graphs is left for future work.

Analysis of the results As shown in Tab. 3 our approach can achieve a large improvement on the multiview registration tasks when compared to the baselines. Not only are the initial pairwise relative transformation parameters estimated using our approach more accurate than the ones of FGR [72], but they can also be further improved in the subsequent iterations. This clearly confirms the benefit of the feed-back loop of our algorithm. Furthermore even when directly considering all input edges our approach still proves dominant, even when considering the results of the scenario "Good" for our competitors. More qualitative results of the multiview registration evaluation, including the failure cases, are available in the supp. material.

Computational complexity Low computational costs of pairwise and multiview registration is important for various fields like augmented reality or robotics. We first compare computation time of our pairwise registration component to RANSAC. In Tab. 2 we report the average time needed to register one fragment pair of the 3DMatch dataset as well

⁴The overlapping regions are defined as parts, where after transformation, the points are less than 0.2m away from the other point cloud. [36]

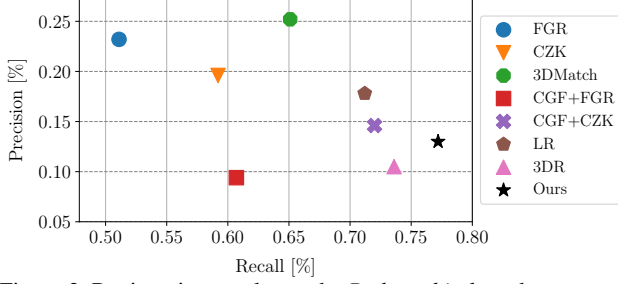


Figure 3. Registration results on the *Redwood indoor* dataset.

as one whole scene. All timings were performed on a standalone computer with Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz, GeForce GTX 1080, and 32 GB RAM. Average time of performing softNN for a fragment pair is about 0.1s, which is a approximately four times faster than traditional nearest neighbor search (implemented using scikit-learn [49]). An even larger speedup (about 23 times) is gained in the model estimation stage, where our approach requires a single forward pass (constant time) compared to up to 50000 iterations of RANSAC when the inlier ratio is 5% and the desired confidence 0.99⁵. This results in an overall run-time of about 80s for our entire multiview approach (including the feature extraction and transformation synchronization) for the *Kitchen* scene with 1770 fragment pairs. In contrast, feature extraction and pairwise estimation of transformation parameters with RANSAC takes > 1100s. This clearly shows the efficiency of our method, being > 13 times faster to compute (for a scene with 60 fragments).

5.3. Ablation study

To get a better intuition how much the individual novelties in our approach contribute to the final performance, we carry out an ablation study on the *ScanNet* [18] dataset. In particular, we analyze the proposed edge pruning scheme based on the confidence estimation block and *Cauchy* function as well as the impact of the iterative refinement of the relative transformation parameters.⁶ The results of the ablation study are presented in Fig. 4.

Benefit from the iterative refinement We motivate the iterative refinement of the transformation parameters that are input to the *Transf-Sync* layer with a notion that the weights and residuals provide additional cues for their estimation. Results in Fig. 4 confirm this assumption. The input relative parameters in the 4-th iteration are approximately 2 percentage points better than the initial estimate. On the other hand, Fig. 4 shows that at the high presence of

⁵We use the CPU-based RANSAC implementation that is provided in the original evaluation code of 3DMatch dataset [70].

⁶Additional results of the ablation study are included in the supp. material.

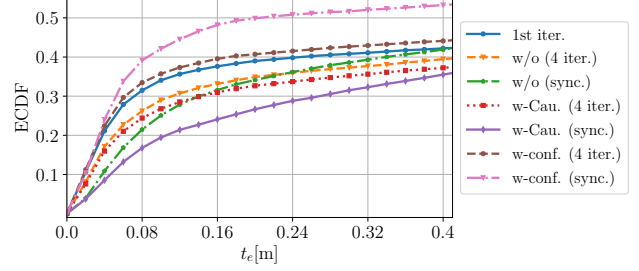


Figure 4. Ablation study on the *ScanNet* dataset.

outliers or inefficient edge pruning (see e.g., the results w/o edge pruning) the weights and the residuals actually provide a negative bias and worsen the results.

Edge pruning scheme There are several possible ways to implement the pruning of the presumable outlier edges. In our experiments we prune the edges based on the output of the confidence estimation block (w-conf.). Other options are to realize this step using the global confidence, i.e. the Cauchy weights defined in (14) (w-Cau.) or not performing this at all (w/o). Fig. 4 clearly shows the advantage of using our confidence estimation block (gain of more than 20 percentage points). Even more, due to preserving a large amount of outliers, alternative approaches perform even worse than the pairwise registration.

6. Conclusions

We have introduced an end-to-end learnable, multiview point cloud registration algorithm. Our method departs from the common two-stage approach and directly learns to register all views in a globally consistent manner. We augment the 3D descriptor FCGF [16] by a soft correspondence layer that pairs all the scans to compute initial matches, which are fed to a differentiable pairwise registration block resulting in transformation parameters as well as weights. A pose graph is constructed and a novel, differentiable iterative transformation synchronization layer globally refines weights and transformations. Experimental evaluation on common benchmark datasets show that our method outperforms state-of-the-art by more than 25 percentage points on average regarding the rotation error statistics. Moreover, our approach is > 13 times faster than RANSAC-based methods in a multiview setting of 60 scans, and generalizes better to new scenes (≈ 4 percentage points higher recall on Redwood indoor compared to state-of-the-art).

Acknowledgements. This work is partially supported by Stanford-Ford Alliance, NSF grant IIS-1763268, Vannevar Bush Faculty Fellowship, Samsung GRO program and the Stanford SAIL Toyota Research Center. We thank NVIDIA Corp. for providing the GPUs used in this work.

References

- [1] Dror Aiger, Niloy J Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM transactions on graphics (TOG)*, number 3, 2008. 2
- [2] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, 2012. 1, 2, 3, 11
- [3] Federica Arrigoni, Luca Magri, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *IEEE International Conference on 3D Vision (3DV)*, pages 491–498, 2014. 1, 2, 3, 11
- [4] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral synchronization of multiple views in se(3). *SIAM Journal on Imaging Sciences*, 9(4):1963–1990, 2016. 1, 3, 5, 7, 13, 14
- [5] Florian Bernard, Johan Thunberg, Peter Gemmar, Frank Hertel, Andreas Husch, and Jorge Goncalves. A solution for multi-alignment by transformation synchronisation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2169, 2015. 1, 2, 3
- [6] PJ Besl and Neil D McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992. 2
- [7] Uttaran Bhattacharya and Venu Madhav Govindu. Efficient and robust registration on the 3d special euclidean group. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 3
- [8] Tolga Birdal and Slobodan Ilic. Point pair features based object detection and pose estimation revisited. In *IEEE International Conference on 3D Vision (3DV)*, 2015. 2
- [9] Tolga Birdal and Slobodan Ilic. Cad priors for accurate and flexible instance reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2
- [10] Tolga Birdal and Umut Simsekli. Probabilistic permutation synchronization using the riemannian structure of the birkhoff polytope. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11105–11116, 2019. 2
- [11] Tolga Birdal, Umut Simsekli, Mustafa Onur Eken, and Slobodan Ilic. Bayesian pose graph optimization via bingham distributions and tempered geodesic mcmc. In *Advances in Neural Information Processing Systems (NIPS)*, pages 308–319, 2018. 1, 3
- [12] A Chatterjee and VM Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2018. 7
- [13] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–528, 2013. 5, 7
- [14] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6, 7
- [15] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019. 4, 12
- [16] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 8958–8966, 2019. 2, 4, 6, 8, 12
- [17] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 864–872, 2015. 2
- [18] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 6, 7, 8, 13, 14
- [19] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *European conference on computer vision (ECCV)*, 2018. 1
- [20] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *European Conference on Computer Vision (ECCV)*, pages 602–618, 2018. 2, 4
- [21] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 195–205, 2018. 1, 2, 4, 6
- [22] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 3d local features for direct pairwise registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6, 7
- [23] Li Ding and Chen Feng. DeepMapping: Unsupervised map estimation from multiple point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8650–8659, 2019. 2
- [24] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 998–1005, 2010. 2
- [25] Simone Fantoni, Umberto Castellani, and Andrea Fusiello. Accurate and automatic alignment of range surfaces. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 73–80. IEEE, 2012. 2
- [26] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 2
- [27] A. Flint, A. Dick, and A. van den Hangel. Thrift: Local 3D structure recognition. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, 2007. 1
- [28] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 4, 6

- [29] Zan Gojcic, Caifa Zhou, and Andreas Wieser. Robust point-wise correspondences for point cloud based deformation monitoring of natural scenes. In *4th Joint International Symposium on Deformation Monitoring (JISDM)*, 2019. 5
- [30] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 684–691, 2004. 1, 3
- [31] Johannes Groß, Aljoša Ošep, and Bastian Leibe. Alignnet-3d: Fast point cloud registration of partially observed objects. In *2019 International Conference on 3D Vision (3DV)*, pages 623–632. IEEE, 2019. 2
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 12
- [33] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. In *European conference on computer vision (ECCV)*, pages 834–848, 2016. 2
- [34] Paul W. Holland and Roy E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827, 1977. 5
- [35] Xiangru Huang, Zhenxiao Liang, Chandrajit Bajaj, and Qixing Huang. Translation synchronization via truncated least squares. In *Advances in neural information processing systems (NIPS)*, pages 1459–1468, 2017. 3, 5
- [36] Xiangru Huang, Zhenxiao Liang, Xiaowei Zhou, Yao Xie, Leonidas J Guibas, and Qixing Huang. Learning transformation synchronization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8082–8091, 2019. 2, 3, 7, 12
- [37] Daniel F Huber and Martial Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7):637–650, 2003. 2
- [38] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 1999. 1
- [39] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 4, 6, 7
- [40] Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. In *International Conference on Learning Representations 2015*, 2015. 6, 12
- [41] Simon Korman and Roei Litman. Latent ransac. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2018. 2, 6
- [42] Hongdong Li and Richard Hartley. The 3D-3D registration problem revisited. In *International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 2
- [43] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcv: An end-to-end deep neural network for point cloud registration. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2
- [44] Eleonora Maset, Federica Arrigoni, and Andrea Fusiello. Practical and efficient multi-view matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4568–4576, 2017. 1, 2
- [45] Nicolas Mellado, Dror Aiger, and Niloy J Mitra. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, 2014. 2
- [46] Ajmal S Mian, Mohammed Bennamoun, and Robyn Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1584–1601, 2006. 2
- [47] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2666–2674, 2018. 5
- [48] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 6
- [49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 8
- [50] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1087–1098, 2018. 4
- [51] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 12
- [52] T. Rabbani, S. Dijkman, F. van den Heuvel, and G. Vosselman. An integrated approach for modelling and global registration of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61:355–370, 2007. 1
- [53] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: a universal framework for random sample consensus. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 2012. 2
- [54] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *European Conference on Computer Vision (ECCV)*, pages 284–299, 2018. 5
- [55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 12
- [56] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009. 1, 2
- [57] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1(1), 2017. 3
- [58] Pascal Theiler, Jan D. Wegner, and Konrad Schindler. Globally consistent registration of terrestrial laser scans via graph optimization. *ISPRS Journal of Photogrammetry and Remote Sensing*, 109:126–136, 2015. 1, 2

- [59] Pascal Willy Theiler, Jan Dirk Wegner, and Konrad Schindler. Keypoint-based 4-points congruent sets—automated marker-less registration of laser scans. *ISPRS journal of photogrammetry and remote sensing*, 2014. **2**
- [60] 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*, 2010. **2**
- [61] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision (ECCV)*, 2010. **1, 2**
- [62] Philip HS Torr and David W Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International journal of computer vision*, 24(3):271–300, 1997. **3**
- [63] Andrea Torsello, Emanuele Rodola, and Andrea Albarelli. Multiview registration via graph diffusion of dual quaternions. In *CVPR 2011*, pages 2441–2448. IEEE, 2011. **1, 2**
- [64] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. **12**
- [65] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3523–3532, October 2019. **1, 2**
- [66] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 38(11):2241–2254, 2015. **2**
- [67] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *European Conference on Computer Vision*, 2018. **1, 2**
- [68] Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent Lepetit. Learning to assign orientations to feature points. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. **5**
- [69] B. Zeisl, K. Köser, and M. Pollefeys. Automatic registration of rgb-d scans via salient directions. In *IEEE International Conference on Computer Vision*, pages 2808–2815, 2013. **1**
- [70] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. **1, 2, 6, 8, 12**
- [71] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Honggen Liao. Learning two-view correspondences and geometry using order-aware network. In *International Conference on Computer Vision (ICCV)*, 2019. **5, 12**
- [72] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision (ECCV)*, pages 766–782, 2016. **2, 3, 6, 7**
- [73] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global sfm by distributed motion averaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2018. **2**

A. Supplementary Material

B. Supplementary material

In this supplementary material, we provide additional information about the proposed algorithm (Sec. **B.1-B.2** and Alg. **1**), network architectures and training configurations (Sec. **B.3**), an extended ablation study (Sec. **B.5**) as well as additional visualizations (Sec. **B.7**). The source code and pretrained models are publicly available under https://github.com/zgojcic/3D_multiview_reg.

B.1. Closed-form solution of Eq. 4.

For the sake of completeness we summarize the closed-form differentiable solution of the weighted least square pairwise registration problem

$$\hat{\mathbf{R}}_{ij}, \hat{\mathbf{t}}_{ij} = \arg \min_{\mathbf{R}_{ij}, \mathbf{t}_{ij}} \sum_{l=1}^N w_l \|\mathbf{R}_{ij} \mathbf{p}_l + \mathbf{t}_{ij} - \mathbf{q}_l\|^2. \quad (22)$$

Let $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$

$$\bar{\mathbf{p}} := \frac{\sum_{l=1}^{N_P} w_l \mathbf{p}_l}{\sum_{l=1}^{N_P} w_l}, \quad \bar{\mathbf{q}} := \frac{\sum_{l=1}^{N_Q} w_l \mathbf{q}_l}{\sum_{l=1}^{N_Q} w_l} \quad (23)$$

denote weighted centroids of point clouds $\mathbf{P} \in \mathbb{R}^{N \times 3}$ and $\mathbf{Q} \in \mathbb{R}^{N \times 3}$, respectively. The centered point coordinates can then be computed as

$$\tilde{\mathbf{p}}_l := \mathbf{p}_l - \bar{\mathbf{p}}, \quad \tilde{\mathbf{q}}_l := \mathbf{q}_l - \bar{\mathbf{q}}, \quad l = 1, \dots, N \quad (24)$$

Arranging the centered points back to the matrix forms $\tilde{\mathbf{P}} \in \mathbb{R}^{N \times 3}$ and $\tilde{\mathbf{Q}} \in \mathbb{R}^{N \times 3}$, a weighted covariance matrix $\mathbf{S} \in \mathbb{R}^{3 \times 3}$ can be computed as

$$\mathbf{S} = \tilde{\mathbf{P}}^T \mathbf{W} \tilde{\mathbf{Q}} \quad (25)$$

where $\mathbf{W} = \text{diag}(w_1, \dots, w_N)$. Considering the singular value decomposition $\mathbf{S} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$ the solution to Eq. 22 is given by

$$\hat{\mathbf{R}}_{ij} = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V} \mathbf{U}^T) \end{bmatrix} \mathbf{U}^T \quad (26)$$

where $\det(\cdot)$ denotes computing the determinant and is used here to avoid creating a reflection matrix. Finally, $\hat{\mathbf{t}}_{ij}$ is computed as

$$\hat{\mathbf{t}}_{ij} = \bar{\mathbf{q}} - \hat{\mathbf{R}}_{ij} \bar{\mathbf{p}} \quad (27)$$

B.2. Closed-form solution of Eq. 5 and 6

In this section we summarize the closed form solutions to Eq. 5 and 6 from the main paper describing the rotation and translation synchronization, respectively.

The least squares formulation of the rotation synchronization problem

$$\mathbf{R}_i^* = \arg \min_{\mathbf{R}_i \in SO(3)} \sum_{(i,j) \in \mathcal{E}} c_{ij} \|\hat{\mathbf{R}}_{ij} - \mathbf{R}_i \mathbf{R}_j^T\|_F^2 \quad (28)$$

admits a closed form solution under spectral relaxation as follows [2, 3]. Consider a symmetric matrix $\mathbf{L} \in \mathbb{R}^{3N_S \times 3N_S}$ resembling a block Laplacian matrix, defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (29)$$

where \mathbf{D} is the weighted degree matrix constructed as

$$\mathbf{D} = \begin{bmatrix} \mathbf{I}_3 \sum_i c_{i1} & & & \\ & \mathbf{I}_3 \sum_i c_{i2} & & \\ & & \ddots & \\ & & & \mathbf{I}_3 \sum_i c_{iN_S} \end{bmatrix} \quad (30)$$

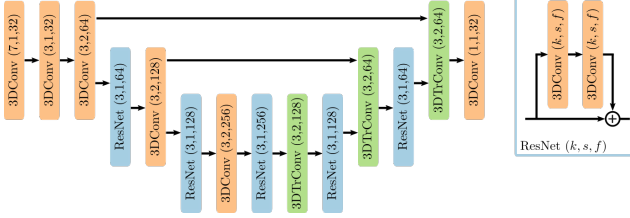


Figure 5. Network architecture of the FCGF [16] feature descriptor. Each convolutional layer (except the last one) is followed by batch normalization and ReLU activation function. The numbers in parentheses denote kernel size, stride, and the number of kernels, respectively.

and \mathbf{A} is a block matrix of the relative rotations

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_3 & c_{12}\hat{\mathbf{R}}_{12} & \cdots & c_{1N_S}\hat{\mathbf{R}}_{1N_S} \\ c_{21}\hat{\mathbf{R}}_{21} & \mathbf{0}_3 & \cdots & c_{2N_S}\hat{\mathbf{R}}_{2N_S} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N_S1}\hat{\mathbf{R}}_{N_S1} & c_{N_S2}\hat{\mathbf{R}}_{N_S2} & \cdots & \mathbf{0}_3 \end{bmatrix} \quad (31)$$

where the weights $c_{ij} := \zeta_{\text{init}}(\mathbf{\Gamma})$ represent the confidence in the relative transformation parameters $\hat{\mathbf{M}}_{ij}$ and N_S denotes the number of nodes in the graph. The least squares estimates of the global rotation matrices $\hat{\mathbf{R}}_i^*$ are then given, under relaxed orthonormality and determinant constraints, by the three eigenvectors $\mathbf{v}_i \in \mathbb{R}^{3N_S}$ corresponding to the smallest eigenvalues of \mathbf{L} . Consequently, the nearest rotation matrices under Frobenius norm can be obtained by a projection of the 3×3 submatrices of $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] \in \mathbb{R}^{3N_S \times 3}$ onto the orthonormal matrices and enforcing the determinant $\det(\hat{\mathbf{R}}_i^*) = 1$ to avoid the reflections.

Similarly, the closed-form solution to the least squares formulation of the translation synchronization

$$\mathbf{t}_i^* = \arg \min_{\mathbf{t}_i} \sum_{(i,j) \in \mathcal{E}} c_{ij} \|\hat{\mathbf{R}}_{ij}\mathbf{t}_i + \hat{\mathbf{t}}_{ij} - \mathbf{t}_j\|^2 \quad (32)$$

can be written as [36]

$$\mathbf{t}^* = \mathbf{L}^+ \mathbf{b} \quad (33)$$

where $\mathbf{t}^* = [\mathbf{t}_1^{*T}, \dots, \mathbf{t}_{N_S}^{*T}]^T \in \mathbb{R}^{3N_S}$ and $\mathbf{b} = [\mathbf{b}_1^{*T}, \dots, \mathbf{b}_{N_S}^{*T}]^T \in \mathbb{R}^{3N_S}$ with

$$\mathbf{b}_i := - \sum_{j \in \mathcal{N}(i)} c_{ij} \hat{\mathbf{R}}_{ij}^T \hat{\mathbf{t}}_{ij}. \quad (34)$$

where $\mathcal{N}(i)$ denotes all the neighboring vertices of \mathbf{S}_i in graph \mathcal{G} .

B.3. Network architecture and training details

This section describes the network architecture as well as the training details of the FCGF [16] feature descriptor (Sec. B.3.1) and the proposed registration block (Sec. ??). Both networks are implemented in Pytorch and pretrained using the *3DMatch* dataset [70].

B.3.1 FCGF local feature descriptor

Network architecture The FCGF [16] feature descriptor operates on sparse tensors that represent a point cloud in form of a set of unique coordinates \mathbf{C} and their associated features \mathbf{F}

$$\mathbf{C} = \begin{bmatrix} x_1 & y_1 & z_1 & b_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & z_N & b_N \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} \quad (35)$$

where x_i, y_i, z_i are the coordinates of the i -th point in the point cloud and f_i is the associated feature (in our case simply 1). FCGF is implemented

using the Minkowski Engine, an auto-differentiation library, which provides support for sparse convolutions and implements all essential deep learning layers [15]. We adopt the original, fully convolutional network design of FCGF that is depicted in Fig. 5. It has a U-Net structure [55] and utilizes skip connections and ResNet blocks [32] to extract the per-point 32 dim feature descriptors. To obtain the unique coordinates \mathbf{C} , we use a GPU implementation of the voxel grid downsampling [15] with the voxel size $v := 2.5$ cm.

Training details We again follow [16] and pre-train FCGF for 100 epochs using the point cloud fragments from the *3DMatch* dataset [70]. We optimize the parameters of the network using stochastic gradient descent with a batch size 4 and an initial learning rate of 0.1 combined with an exponential decay with $\gamma = 0.99$. To introduce rotation invariance of the descriptors we perform a data augmentation by randomly rotating each of the fragments along an arbitrary direction, by a different rotation, sampled from the $[0^\circ, 360^\circ)$ interval. The sampling of the positive and negative examples follows the procedure proposed in [16].

B.3.2 Registration block

Network architecture The architecture of the registration block (same for $\psi_{\text{init}}(\cdot)$ and $\psi_{\text{iter}}(\cdot)$)⁷ follows [71] and is based on the PointNet-like architecture [51] where each of the fully connected layers (\mathbf{P} in Fig. 6) operates on individual correspondences. The local context is then aggregated using the instance normalization layers [64] defined as

$$\mathbf{y}_i^l = \frac{\mathbf{x}_i^l - \boldsymbol{\mu}^l}{\boldsymbol{\sigma}^l} \quad (36)$$

where \mathbf{x}_i^l is the output of the layer l and $\boldsymbol{\mu}^l$ and $\boldsymbol{\sigma}^l$ are per dimension mean value and standard deviation, respectively. Opposed to the more commonly used batch normalization, instance normalization operates on individual training examples and not on the whole batch. Additionally, to reinforce the local context, the order-aware blocks [71] are used to map the correspondences to clusters using the learned soft pooling $\mathbf{S}_{\text{pool}} \in \mathbb{R}^{N_c \times M_c}$ and unpooling $\mathbf{S}_{\text{unpool}} \in \mathbb{R}^{N_c \times M_c}$ operators as

$$\mathbf{X}_{k+1} = \mathbf{S}_{\text{pool}}^T \mathbf{X}_k \quad \text{and} \quad \mathbf{X}'_k = \mathbf{S}_{\text{unpool}} \mathbf{X}'_{k+1} \quad (37)$$

where N_c is the number of correspondences and M_c is the number of clusters. \mathbf{X}_k and \mathbf{X}_{k+1} are the features at the level k (before clustering) and $k+1$ (after clustering), respectively (see Fig. 6). Finally, \mathbf{X}'_{k+1} denotes the output of the last layer in the level $k+1$.

Training details We pre-train the registration blocks using the same fragments from the *3DMatch* dataset. Specifically, we first infer the FCGF descriptors and randomly sample $N_c = 5000$ descriptors per fragment. We use these descriptors to compute the putative correspondences for all fragment pairs (i, j) such that $i \leq j$. Based on the ground truth transformation parameters, we label these correspondences as inliers if the Euclidean distance between the points after the transformation is smaller than 7.5 cm. At the start of the training (first 15000 iterations) we supervise the learning using only the binary cross-entropy loss. Once a meaningful number of correspondences can already be classified correctly we add the transformation loss. We train the network for 500k iterations using Adam [40] optimizer with the initial learning rate of 0.001. We decay the learning rate every 1000 iterations by multiplying it with 0.999. To learn the rotation invariance we perform data augmentation, starting from the 25000th iteration, by randomly sampling an angle from the interval $[0^\circ, n_a \cdot 20^\circ)$ where n_a is initialized with zero and is then increased by 1 every 5000 iteration until the interval becomes $[0^\circ, 360^\circ)$.

⁷For $\psi_{\text{init}}(\cdot)$ the input dimension is increased from 6 to 8 (weights and residuals added).

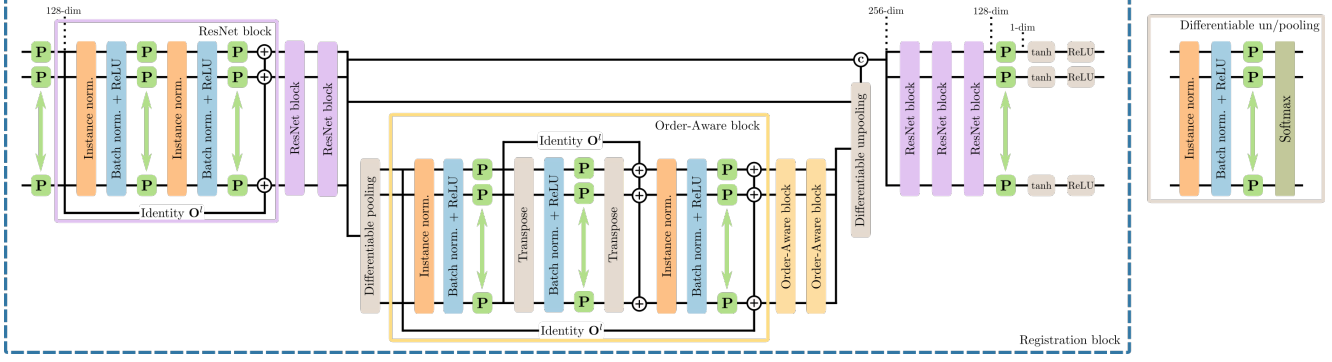


Figure 6. Network architecture of the registration block consists of two main modules: i) a PointNet-like ResNet block with instance normalization, and ii) an order-aware block. For each point cloud pair, putative correspondences are fed into three consecutive ResNet blocks followed by a differentiable pooling layer, which maps the N_c putative correspondences to M_c clusters \mathbf{X}_{k+1} at the level $k+1$. These serve as input to the three order-aware blocks. Their output \mathbf{X}'_{k+1} is fed along with \mathbf{X}_k into the differentiable unpooling layer. The recovered features are then used as input to the remaining three ResNet blocks. The output of the registration block are the scores s_i indicating whether the putative correspondence is an outlier or an inlier. Additionally, the 128-dim features (denoted as $\mathbf{X}^{\text{conf}} := f_{\theta}^{-2}(\cdot)$) before the last perceptron layer \mathbf{P} are used as input to the confidence estimation block.

B.4. Pseudo-code

Alg. 1 shows the pseudo-code of our proposed approach. We iterate $k = 4$ times over the network and transformation synchronization (i.e. *Transf-Sync*) layers and in each of those iterations we execute the *Transf-Sync* layer four times. Our implementation is constructed in a modular way (each part can be run on its own) and can accept a varying number of input point clouds with or without the connectivity information.

B.5. Extended ablation study

We extend the ablation study presented in the main paper, by analyzing the impact of edge pruning based on the local confidence (i.e. the output of the confidence estimation block) (Sec. B.5.1) and of the weighting scheme (Sec. B.6) on the angular and translation errors. The ablation study is performed on the point cloud fragments of the *ScanNet* dataset [18].

B.5.1 Impact of the edge pruning threshold

Results depicted in Fig. 7 show that the threshold value used for edge pruning has little impact on the angular and translation errors as long as it is larger than 0.2.

B.6. Impact of the harmonic mean weighting scheme

In this work, we have introduced a scheme for combining the local and global confidence using the harmonic mean (HM). In the following, we perform the analysis of this proposal and compare its performance to established methods based only on global information [4]. To this end, we again consider the scenario “Ours (Good)” as the input graph connectivity information. We compare the results of the proposed scheme (HM) to SE3 EIG [4], which proposes using the Cauchy function for computing the global edge confidence [4]. Note, we use the same pairwise transformation parameters, estimated using the method proposed herein, for all methods.

Without edge pruning It turns out that combining the local and global evidence about the graph connectivity is essential to achieve good performance. In fact, merely relying on local confidence estimates without HM weighting (denoted as ours; green) in Fig. 8) the *Transf-Sync* is unable to recover global transformations from the given graph connectivity evidence that is very noisy. Introducing the HM weighting scheme allows

Algorithm 1 Pseudo-code of the proposed approach

Input: a set of potentially overlapping scans $\{\mathbf{S}_i\}_{i=1}^{N_S}$
Output: globally optimized poses $\{\mathbf{M}_i^*\}_{i=1}^{N_S}$
Compute the pairwise transformations
for each pair of scans $\mathbf{S}_i, \mathbf{S}_j \subset \mathcal{S}, i \neq j$ do
 # find the putative correspondences using $\phi(\cdot)$
 - $\mathbf{X}_{ij} = \text{cat}([\mathbf{S}_i, \phi(\mathbf{S}_i, \mathbf{S}_j)]) \in \mathbb{R}^{N_{S_i} \times 6}$
 # compute the weights $\mathbf{w}_{ij} \in \mathbb{R}^{N_{S_i}}$ using $\psi_{init}(\cdot)$
 - $\mathbf{w}_{ij} = \psi_{init}(\mathbf{X}_{ij}) \in \mathbb{R}^{N_{S_i}}$
 - calculate $\mathbf{R}_{ij}, \mathbf{t}_{ij}$ using SVD according to (4)
Iterative network for transformation synchronization
 $\mathbf{X}_{ij}^{(0)} \leftarrow \mathbf{X}_{ij}, \mathbf{w}_{ij}^{(0)} \leftarrow \mathbf{w}_{ij}, \mathbf{r}_{ij}^{(0)} \leftarrow \mathbf{r}_{ij}$
for $k = 1, 2, \dots, \text{max_iters}$ do
 for each pairwise output from ψ_{init} do
 - $\mathbf{R}_{ij}^{(k)}, \mathbf{t}_{ij}^{(k)}, \mathbf{w}_{ij}^{(k)} = \psi_{iter}([\mathbf{X}_{ij}^{(k-1)}, \mathbf{w}_{ij}^{(k-1)}, \mathbf{r}_{ij}^{(k-1)}])$
 - estimate $\text{local}\{c_{ij}^{(k)}\}$ using (16)
 - Gather the pairwise estimation as $\mathbf{R}^{(k)}, \mathbf{t}^{(k)}, \mathbf{c}^{(k)}$
 # Build the graph and perform the synchronization
 if $k = 1$ then
 - $\mathbf{c}^{(k)} := \text{local}\{\mathbf{c}^{(k)}\}$
 else
 - $\mathbf{c}^{(k)} := f_{HM}(\text{local}\{\mathbf{c}^{(k)}\}, \text{global}\{\mathbf{c}^{(k-1)}\})$
 - $\mathbf{R}^{*(k)}, \mathbf{t}^{*(k)} = \text{Transf-Sync}(\mathbf{R}^{(k)}, \mathbf{t}^{(k)}, \mathbf{c}^{(k)})$
 # update step
 for each pair of scans $\mathbf{S}_i, \mathbf{S}_j \subset \mathcal{S}, i \neq j$ do
 - $\mathbf{X}_{ij}^{(k+1)} = \text{cat}([\mathbf{S}_i, \mathbf{M}_{ij}^{*(k)}] \otimes \phi(\mathbf{S}_i, \mathbf{S}_j))$
 - $\mathbf{w}_{ij}^{(k+1)} = \mathbf{w}_{ij}^{(k)}$
 - $\mathbf{r}_{ij}^{(k+1)} = \|\mathbf{S}_i - \mathbf{M}_{ij}^{*(k)}] \otimes \phi(\mathbf{S}_i, \mathbf{S}_j)\|_2$

us to reduce the impact of noisy graph connectivity built solely using local confidence and can significantly improve performance after *Transf-Sync*

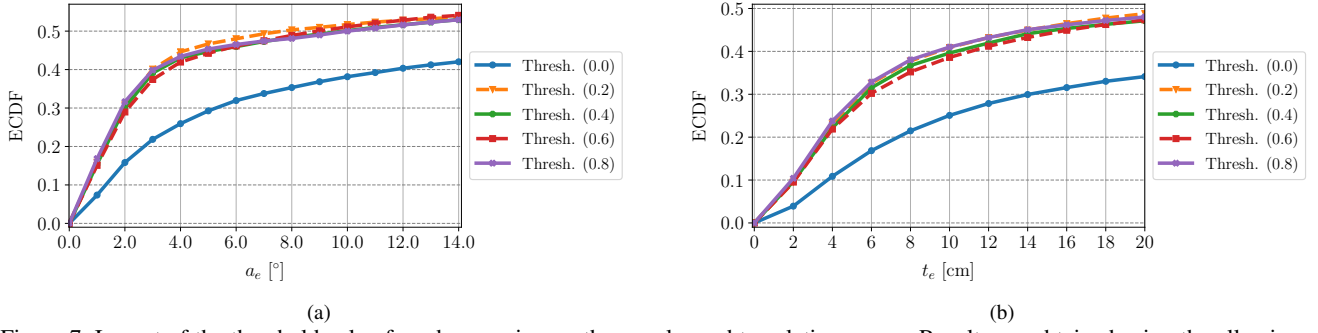


Figure 7. Impact of the threshold value for edge pruning on the angular and translation errors. Results are obtained using the all pairs as input graph on *ScanNet* dataset [18]. (a) angular error and (b) translation error.

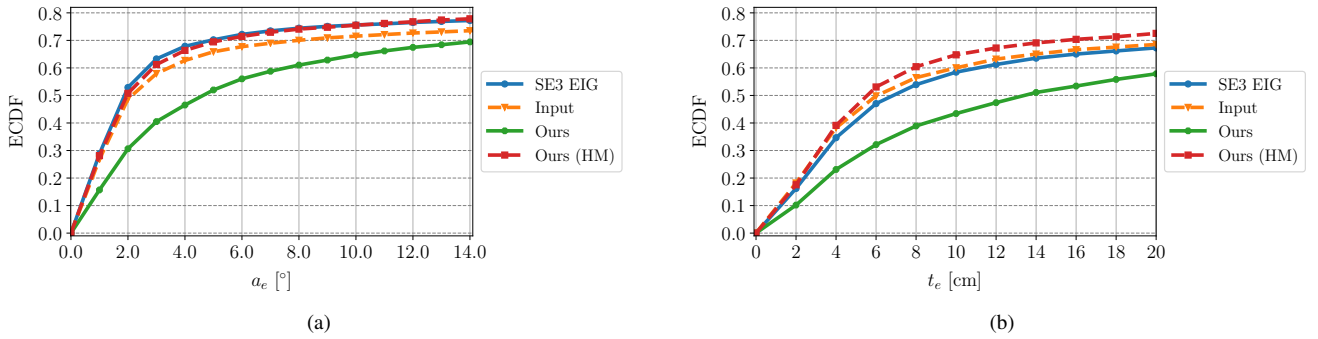


Figure 8. Impact of the weighting scheme without edge cutting on the angular and translation errors. (a) angular and (b) translation errors.

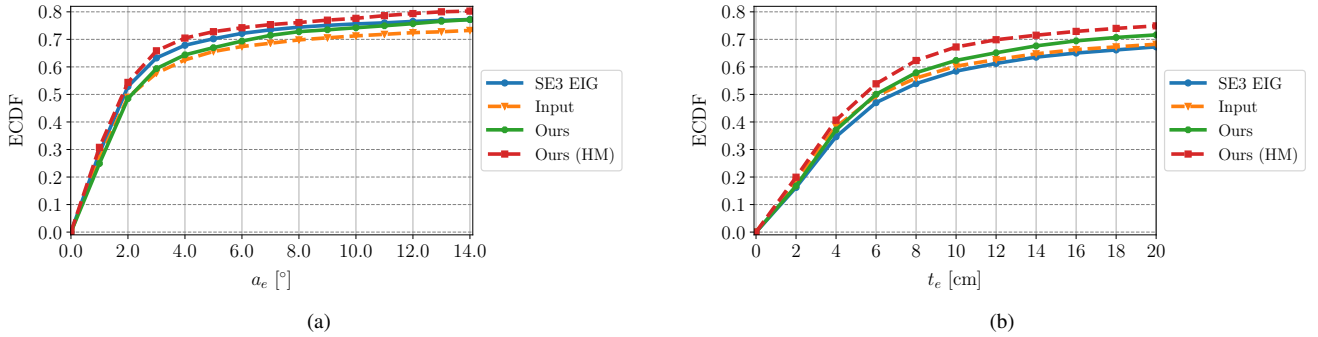


Figure 9. Impact of the weighting scheme combined with edge cutting, on the angular and translation errors. (a) angular error and (b) translation error.

block, which in turn enables us to outperform the *SE3 EIG*.

With edge pruning Fig. 9 shows that pruning the edges can help coping with noisy input graph connectivity built from the pairwise input. In principal, suppression of the edges with low confidence results in discarding the outliers that corrupt the l_2 solution and as a result improves the performance of the *Transf-Sync* block.

B.7. Qualitative results

We provide some additional qualitative results in form of success and failure cases on selected scenes of *3DMatch* (Fig. 10 and 11) and *ScanNet* (Fig. 12 and 13) datasets. Specifically, we compare the results of our whole pipeline *Ours (After Sync.)* to the results of *SE3 EIG* [4], pairwise registration results of our method from the first iteration *Ours (1st*

iter.), and pairwise registration results of our method from the fourth iteration *Ours (4th iter.)*. Both global methods (*Ours (After Sync.)* and *SE3 EIG*) use transformation parameters estimated by our proposed pairwise registration algorithm as input to the transformation synchronization. The failure cases of our method predominantly occur on point clouds with low level of structure (planar areas in Fig. 11 bottom) or high level of symmetry and repetitive structures (Fig. 13 top and bottom, respectively).

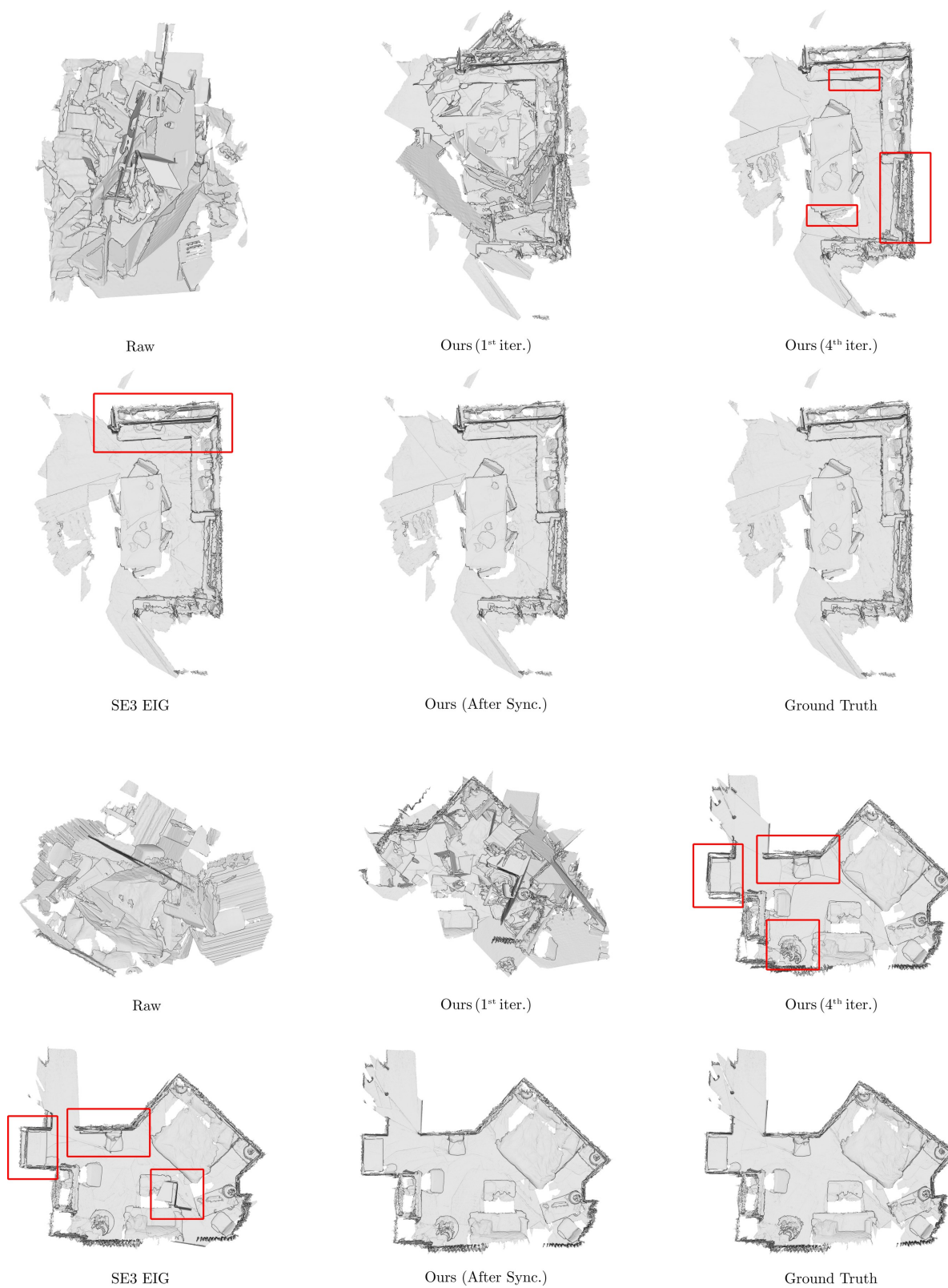


Figure 10. Selected **success cases** of our method on *3DMatch* dataset. Top: **Kitchen** and bottom: **Hotel 1**. Red rectangles highlight interesting areas with subtle changes.

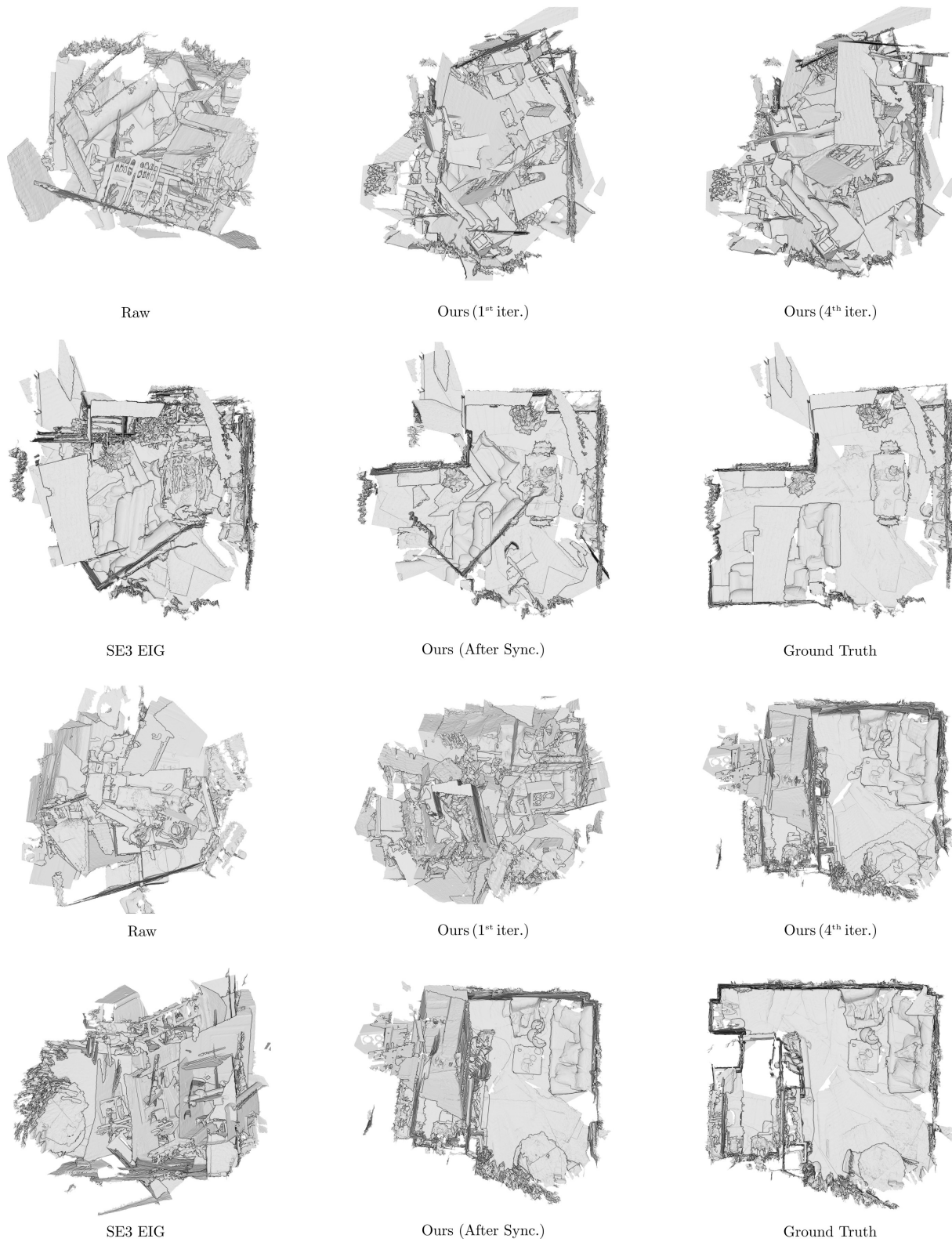
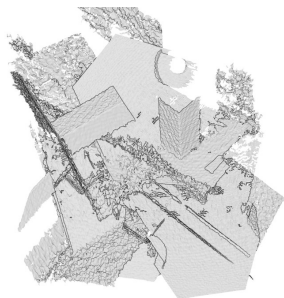


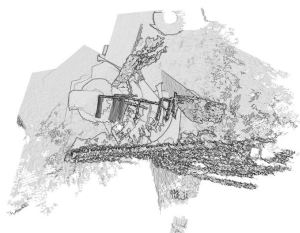
Figure 11. Selected **failure cases** of our method on *3DMatch* dataset. Top: **Home 1** and bottom: **Home 2**. Note that our method still provides qualitatively better results than state-of-the-art.



Figure 12. Selected **success cases** of our method on *ScanNet* dataset. Top: **scene0057_01** and bottom: **scene0309_00**. Red rectangles highlight interesting areas with subtle changes.



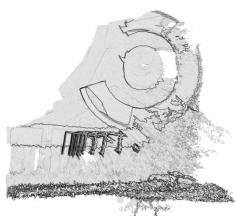
Raw



Ours (1st iter.)



Ours (4th iter.)



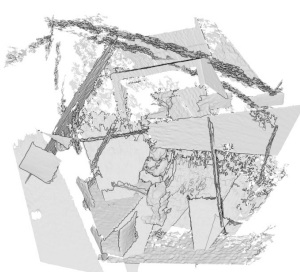
SE3 EIG



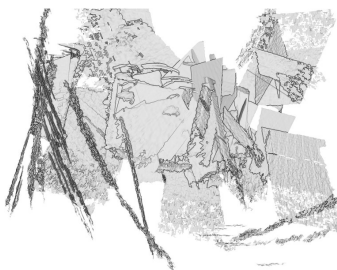
Ours (After Sync.)



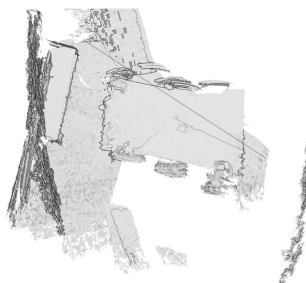
Ground Truth



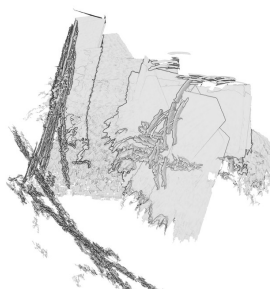
Raw



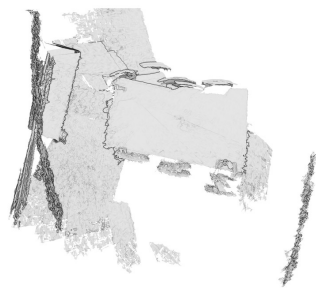
Ours (1st iter.)



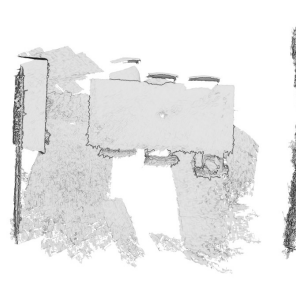
Ours (4th iter.)



SE3 EIG



Ours (After Sync.)



Ground Truth

Figure 13. Selected **failure** cases of our method on *ScanNet* dataset. Top: **scene0334_02** and bottom: **scene0493_01**. Note that our method still provides qualitatively better results than state-of-the-art.