Learning Transformation Synchronization

Xiangru Huang
UT Austin

Zhenxiao Liang
UT Austin

Xiaowei Zhou
Zhejiang University*

Yao Xie
Georgia Tech

Leonidas Guibas
Facebook AI Research, Stanford University

Qixing Huang†
UT Austin

Abstract

Reconstructing the 3D model of a physical object typically requires us to align the depth scans obtained from different camera poses into the same coordinate system. Solutions to this global alignment problem usually proceed in two steps. The first step estimates relative transformations between pairs of scans using an off-the-shelf technique. Due to limited information presented between pairs of scans, the resulting relative transformations are generally noisy. The second step then jointly optimizes the relative transformations among all input depth scans. A natural constraint used in this step is the cycle-consistency constraint, which allows us to prune incorrect relative transformations by detecting inconsistent cycles. The performance of such approaches, however, heavily relies on the quality of the input relative transformations. Instead of merely using the relative transformations as the input to perform transformation synchronization, we propose to use a neural network to learn the weights associated with each relative transformation. Our approach alternates between transformation synchronization using weighted relative transformations and predicting new weights of the input relative transformations using a neural network. We demonstrate the usefulness of this approach across a wide range of datasets.

1. Introduction

Transformation synchronization, i.e., estimating consistent rigid transformations across a collection of images or depth scans, is a fundamental problem in various computer vision applications, including multi-view structure from motion [11, 37, 48, 45], geometry reconstruction from depth scans [27, 15], image editing via solving jigsaw puzzles [14], simultaneous localization and mapping [10], and reassembling fractured surfaces [22], to name just a few. A common approach to transformation synchronization proceeds in two phases. The first phase establishes the relative rigid transformations between pairs of objects in isolation. Due to incomplete information presented in isolated pairs, the estimated relative transformations are usually quite noisy. The second phase improves the relative transformations by jointly optimizing them across all input objects. This is usually made possible by utilizing the so-called *cycle-consistency* constraint, which states that the composite transformation along every cycle should be the identity transformation, or equivalently, the data matrix that stores pair-wise transformations in blocks is low-rank (c.f. [23]). This cycle-consistency constraint allows us to jointly improve relative transformations by either detecting inconsistent cycles [14, 36] or performing low-rank matrix recovery [23, 47, 39, 7, 9].

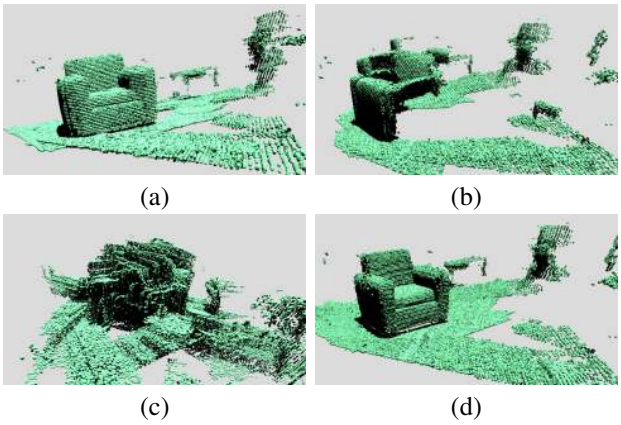

Figure 1: Reconstruction results from 30 RGBD images of an indoor environment using different transformation synchronization methods. (a) Our approach. (b) Rotation Averaging [12]. (c) Geometric Registration[15]. (d) Ground Truth.

However, the success of existing transformation synchronization [47, 11, 3, 26] and more general map synchronization [23, 39, 38, 13, 42, 26] techniques heavily depends on the compatibility between the loss function and the noise pattern of the input data. For example, approaches based on robust norms (e.g., L1 [23, 13]) can tolerate either a constant fraction of adversarial noise (c.f.[23, 26])

*Xiaowei Zhou is affiliated with the StateKey Lab of CAD&CG and the ZJU-SenseTime Joint Lab of 3D Vision.

†huangqx@cs.utexas.edu

or a sub-linear outlier ratio when the noise is independent (c.f.[13, 42]). Such assumptions, unfortunately, deviate from many practical settings, where the majority of the input relative transformations may be incorrect (e.g., when the input scans are noisy), and/or the noise pattern in relative transformations is highly correlated (there are a quadratic number of measurements from a linear number of sources). This motivates us to consider the problem of *learning transformation synchronization*, which seeks to learn a suitable loss function that is compatible with the noise pattern of specific datasets.

In this paper, we introduce an approach that formulates transformation synchronization as an end-to-end neural network. Our approach is motivated by reweighted least squares and their application in transformation synchronization (c.f. [11, 3, 15, 26]), where the loss function dictates how we update the weight associated with each input relative transformation during the synchronization process. Specifically, we design a recurrent neural network that reflects this reweighted scheme. By learning the weights from data directly, our approach implicitly captures a suitable loss function for performing transformation synchronization.

We have evaluated the proposed technique on two real datasets: Redwood [16] and ScanNet [17]. Experimental results show that our approach leads to considerable improvements compared to the state-of-the-art transformation synchronization techniques. For example, on Redwood and Scannet, the best combination of existing pairwise matching and transformation synchronization techniques lead to mean angular rotation errors $22.4°$ and $64.4°$, respectively. In contrast, the corresponding statistics of our approach are $6.9°$ and $42.9°$, respectively. We also perform an ablation study to evaluate the effectiveness of our approach.

Code is publicly available at https://github.com/xiangruhuang/Learning2Sync.

2. Related Works

Existing techniques on transformation synchronization fall into two categories. The first category of methods [27, 22, 49, 36, 52] uses combinatorial optimization to select a subgraph that only contains consistent cycles. The second category of methods [47, 31, 25, 23, 24, 13, 53, 42, 33, 26, 7, 39, 38, 2, 9, 4, 5, 41, 19, 46, 6, 21] can be viewed from the perspective that there is an equivalence between cycle-consistent transformations and the fact that the map collection matrix that stores relative transformations in blocks is semidefinite and/or low-rank (c.f.[23]). These methods formulate transformation synchronization as low-rank matrix recovery, where the input relative transformations are considered noisy measurements of this low-rank matrix. In the literature, people have proposed convex optimization [47, 23, 24, 13], non-convex optimization [11, 53, 33, 26], and spectral techniques [31, 25, 39, 38, 42, 44, 7, 2, 9] for solving various low-rank matrix recovery formulations. Compared with the first category of methods, the second category of methods is computationally more efficient. Moreover, tight exact recovery conditions of many methods have been established.

A message from these exact recovery conditions is that existing methods only work if the fraction of noise in the input relative transformations is below a threshold. The magnitude of this threshold depends on the noise pattern. Existing results either assume adversarial noise [23, 26] or independent random noise [47, 13, 42, 8]. However, as relative transformations are computed between pairs of objects, it follows that these relative transformations are dependent (i.e., between the same source object to different target objects). This means there are a lot of structures in the noise pattern of relative transformations. Our approach addresses this issue by optimizing transformation synchronization techniques to fit the data distribution of a particular dataset. To best of our knowledge, this work is the first to apply supervised learning to the problem of transformation synchronization.

Our approach is also relevant to utilizing recurrent neural networks for solving the pairwise matching problem. Recent examples include learning correspondences between pairs of images [35], predicting the fundamental matrix between two different images of the same underlying environment [40], and computing a dense image flow between an image pair [30]. In contrast, we study a different problem of transformation synchronization in this paper. In particular, our weighting module leverages problem specific features (e.g., eigen-gap) for determining the weights associated with relative transformations. Learning transformation synchronization also poses great challenges in making the network trainable end-to-end.

3. Problem Statement and Approach Overview

In this section, we describe the problem statement of transformation synchronization (Section 3.1) and present an overview of our approach (Section 3.2).

3.1. Problem Statement

Consider n input scans $\mathcal{S} = \{S_i, 1 \leq i \leq n\}$ capturing the same underlying object/scene from different camera poses. Let Σ_i denote the local coordinate system associated with S_i. The input to transformation synchronization can be described as a model graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ [28]. Each edge $(i, j) \in \mathcal{E}$ of the model graph is associated with a relative transformation $T_{ij}^{in} = (R_{ij}^{in}, \boldsymbol{t}_{ij}^{in}) \in \mathbb{R}^{3\times 4}$, where $R_{ij}^{in} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{t}_{ij}^{in} \in \mathbb{R}^3$ are rotational and translational components of T_{ij}^{in}, respectively. T_{ij}^{in} is usually precomputed using an off-the-shelf algorithm (e.g., [34, 50]). For simplicity, we impose the assumption that $(i, j) \in \mathcal{E}$ if and only if (i) $(j, i) \in \mathcal{E}$, and (ii) their associated transformations are compatible, i.e.,

$$R_{ji}^{in} = R_{ij}^{in^T}, \quad \boldsymbol{t}_{ji}^{in} = -R_{ij}^{in^T}\boldsymbol{t}_{ij}^{in}.$$

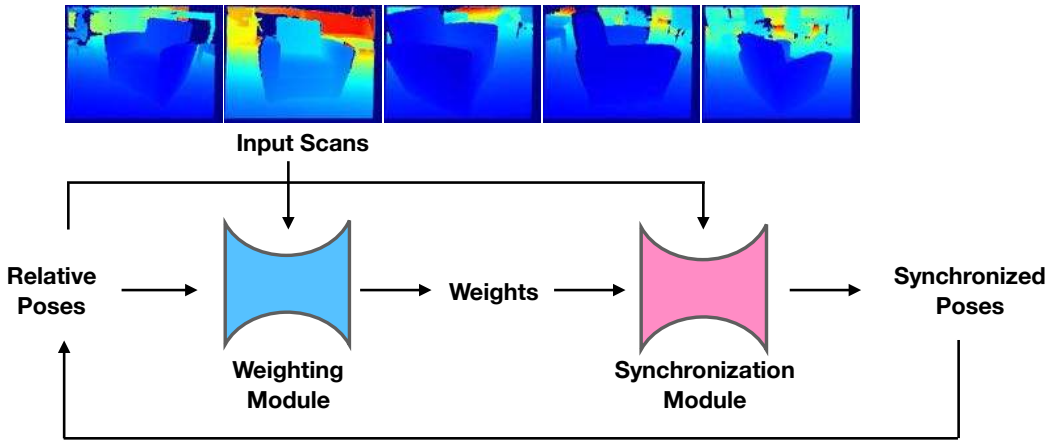

Figure 2: Illustration of our network design.

It is expected that many of these relative transformations are incorrect, due to limited information presented between pairs of scans and limitations of the off-the-shelf method being used. The goal of transformation synchronization is to recover the absolute pose $T_i = (R_i, \boldsymbol{t}_i) \in \mathbb{R}^{3\times4}$ of each scan S_i in a world coordinate system Σ. Without losing generality, we assume the world coordinate system is given by $\Sigma := \Sigma_1$. Note that unlike traditional transformation synchronization approaches that merely use T_{ij}^{in} (e.g.,[11, 47, 3]), our approach also incorporates additional information extracted from the input scans $S_i, 1 \leq i \leq n$.

3.2. Approach Overview

Our approach is motivated from iteratively reweighted least squares (or IRLS)[18], which has been applied to transformation synchronization (e.g. [11, 3, 15, 26]). The key idea of IRLS is to maintain an edge weight $w_{ij}, (i, j) \in \mathcal{E}$ for each input transformation T_{ij}^{in} so that the objective function becomes quadratic in the variables, and transformation synchronization admits a closed-form solution. One can then use the closed-form solution to update the edge weights. One way to understand reweighting schemes is that when the weights converged, the reweighted square loss becomes the actual robust loss function that is used to solve the corresponding transformation synchronization problem. In contrast to using a generic weighting scheme, we propose to learn the weighting scheme from data by designing a recurrent network that replicates the reweighted transformation synchronization procedure. By doing so, we implicitly learn a suitable loss function for transformation synchronization.

As illustrated in Figure 2, the proposed recurrent module combines a synchronization layer and a weighting module. At the kth iteration, the synchronization layer takes as input the initial relative transformations $T_{ij}^{in} \in \mathbb{R}^{3\times4}, \forall (i, j) \in \mathcal{E}$ and their associated weights $w_{ij}^{(k)} \in (0, 1)$ and outputs synchronized poses $T_i^{(k)} : \Sigma_i \to \Sigma$ for the input objects $S_i, 1 \leq i \leq n$. Initially, we set $w_{ij}^{(1)} = 1, \forall (i, j) \in \mathcal{E}$. The technical details of the synchronization layer are described in Section 4.1.

The weighting module operates on each object pair in isolation. For each edge $(i, j) \in \mathcal{E}$, the input to the proposed weighting module consists of (1) the input relative transformation T_{ij}^{in}, (2) features extracted from the initial alignment of the two input scans, and (3) a status vector $\boldsymbol{v}^{(k)}$ that collects global signals from the synchronization layer at the kth iteration (e.g., spectral gap). The output is the associated weight $w_{ij}^{(k+1)}$ at the $k + 1$th iteration.

The network is trained end-to-end by penalizing the differences between the ground-truth poses and the output of the last synchronization layer. The technical details of this end-to-end training procedure are described in Section 4.3.

4. Approach

In this section, we introduce the technical details of our learning transformation synchronization approach. In Section 4.1, we introduce details of the synchronization layer. In Section 4.2, we describe the weighting module. Finally, we show how to train the proposed network end-to-end in Section 4.3. Note that the proofs of the propositions introduced in this section are deferred to the supplementary material.

4.1. Synchronization Layer

For simplicity, we ignore the superscripts ^{k} and ^{in} when introducing the synchronization layer. Let $T_{ij} = (R_{ij}, \boldsymbol{t}_{ij})$ and w_{ij} be the input relative transformation and its weights associated with the edge $(i, j) \in \mathcal{E}$. We assume that this weighted graph is connected. The goal of the synchronization layer is to compute the synchronized pose $T_i^\star = (R_i^\star, \boldsymbol{t}_i^\star)$ associated with each scan S_i. Note that a correct relative transformation $T_{ij} = (R_{ij}, \boldsymbol{t}_{ij})$ induces two sepa-

Algorithm 1 Translation Synchronization Layer.

function SYNC($(w_{ij}, T_{ij}), \forall (i,j) \in \mathcal{E}$)
Form the connection Laplacian L and vector $\boldsymbol{b}$;
Compute first 3 eigenvectors U of L;
Perform SVD on blocks of U to obtain $\{R_i^\star, 1 \le i \le n\}$ via (2);
Solve (4) to obtain $\{\boldsymbol{t}_i^\star, 1 \le i \le n\}$;
return $T_i^\star = (R_i^\star, \boldsymbol{t}_i^\star), 1 \le i \le n$;
end function

rate constraints on the rotations $R_i^\star$ and translations $\boldsymbol{t}_i^\star$, respectively:

$$R_{ij} R_i^\star = R_j^\star, \quad R_{ij}\boldsymbol{t}_i^\star + \boldsymbol{t}_{ij} = \boldsymbol{t}_j^\star.$$

We thus perform rotation synchronization and translation synchronization separately.

Rotation synchronization. Our rotation synchronization approach adapts a Laplacian rotation synchronization formulation proposed in the literature [1, 2, 9, 4]. More precisely, we introduce a connection Laplacian $L \in \mathbb{R}^{3n\times 3n}$ [43], whose blocks are given by

$$L_{ij} := \begin{cases} \sum\limits_{j\in\mathcal{N}(i)} w_{ij} I_3 & i = j \\ -w_{ij} R_{ij}^T & (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\mathcal{N}(i)$ collects all neighbor vertices of i in $\mathcal{G}$.

Let $U = (U_1^T, \cdots, U_n^T)^T \in \mathbb{R}^{3n\times 3}$ collect the eigenvectors of L that correspond to the three smallest eigenvalues. We choose the sign of each eigenvector such that $\sum_{i=1}^n \det(U_i) > 0$. To compute the absolute rotations, we first perform singular value decomposition (SVD) on each

$$U_i = V_i \Sigma_i W_i^T.$$

We then output the corresponding absolute rotation estimate as

$$R_i^* = V_i W_i^T \tag{2}$$

It can be shown that when the observation graph is connected and $R_{ij}, (i,j) \in \mathcal{E}$ are exact, then $R_i^*, 1 \le i \le n$ recover the underlying ground-truth solution (c.f.[1, 2, 9, 4]). In Section C.3 of the supplementary material, we present a robust recovery result that $R_i^\star$ approximately recover the underlying ground-truth even when R_{ij} are inexact.

Translation synchronization solves the following least square problem to obtain $\boldsymbol{t}_i$:

$$\underset{\boldsymbol{t}_i, 1\le i\le n}{\text{minimize}} \sum_{(i,j)\in\mathcal{E}} w_{ij} \|R_{ij}\boldsymbol{t}_i + \boldsymbol{t}_{ij} - \boldsymbol{t}_j\|^2 \tag{3}$$

Let $\boldsymbol{t} = (\boldsymbol{t}_1^T, \cdots, \boldsymbol{t}_n^T)^T \in \mathbb{R}^{3n}$ collect the translation components of the synchronized poses in a column vector. Introduce a column vector $\boldsymbol{b} = (\boldsymbol{b}_1^T, \cdots, \boldsymbol{b}_n^T)^T \in \mathbb{R}^{3n}$ where

$$\boldsymbol{b}_i := -\sum_{j\in\mathcal{N}(i)} w_{ij} R_{ij}^T \boldsymbol{t}_{ij}.$$

Then an[1] optimal solution $\boldsymbol{t}^\star$ to (3) is given by

$$\boldsymbol{t}^\star = L^+ \boldsymbol{b}. \tag{4}$$

Similar to the case of rotation synchronization, we can show that when the observation graph is connected, and $R_{ij}, \boldsymbol{t}_{ij}, (i,j) \in \mathcal{E}$ are exact, then $\boldsymbol{t}^\star$ recovers the underlying ground-truth rotations. Section C.4 of the supplementary material presents a robust recovery result for translations.

4.2. Weighting Module

We define the weighting module as the following function:

$$w_{ij}^{(k+1)} \leftarrow \text{Weight}_\theta(S_i, S_j, T_{ij}^{in}, \boldsymbol{s}_{ij}^{(k)}) \tag{5}$$

where the input consists of (i) a pair of scans S_i and S_j, (ii) the input relative transformation T_{ij}^{in} between them, and (iii) a status vector $\boldsymbol{s}_{ij}^{(k)} \in \mathbb{R}^4$. The output of this weighting module is given by the new weight $w_{ij}^{(k+1)}$ at the $k+1$th iteration. With θ we denote the trainable weights of the weighting module. In the following, we first introduce the definition of the status vector $\boldsymbol{s}_{ij}^{(k)}$.

Status vector. The purpose of the status vector $\boldsymbol{s}_{ij}^{(k)}$ is to collect additional signals that are useful for determining the output of the weighting module. Define

$$s_{ij1}^{(k)} := \|R_{ij}^{in} - R_j^{(k)} {R_i^{(k)}}^T\|_{\mathcal{F}}, \tag{6}$$

$$s_{ij2}^{(k)} := \|R_{ij}^{in}\boldsymbol{t}_i^{(k)} + \boldsymbol{t}_{ij}^{in} - \boldsymbol{t}_j^{(k)}\|. \tag{7}$$

$$s_{ij3}^{(k)} := \lambda_4(L^{(k)}) - \lambda_3(L^{(k)}), \tag{8}$$

$$s_{ij4}^{(k)} := \sum_{(i,j)\in\mathcal{E}} w_{ij}^{(k)} \|\boldsymbol{t}_{ij}^{(k)}\|^2 - {\boldsymbol{b}^{(k)}}^T {L^{(k)}}^+ \boldsymbol{b}^{(k)}, \tag{9}$$

Essentially, $s_{ij1}^{(k)}$ and $s_{ij2}^{(k)}$ characterize the difference between current synchronized transformations and the input relative transformations. The motivation for using them comes from the fact that for a standard reweighted scheme for transformation synchronization (c.f. [26]), one simply sets $w_{ij}^{(k+1)} = \rho(s_{ij1}^{(k)}, s_{ij2}^{(k)})$ for a weighting function ρ (c.f. [18]). This scheme can already recover the underlying ground-truth in the presence of a constant fraction of adversarial incorrect relative transformations (Please refer to Section C.7 of the supplementary material for a formal analysis). In contrast, our approach seeks to go beyond this limit by leveraging additional information. The definition of $s_{ij3}^{(k)}$ captures the spectral gap of the connection Laplacian. $s_{ij4}^{(k)}$ equals to the residual of (3). Intuitively, when $s_{ij3}^{(k)}$ is large and $s_{ij4}^{(k)}$ is small, the weighted relative transformations $w_{ij}^{(k)} \cdot T_{ij}^{in}$ will be consistent, from which we can re-

[1] When L is positive semidefinite, then the solution is unique, and (4) gives one optimal solution.

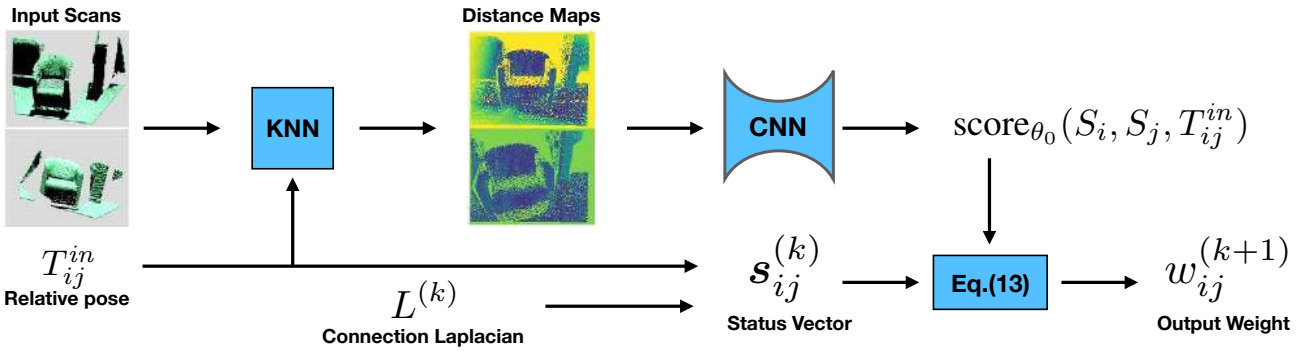

Figure 3: Illustration of network design of the weighting module. We first compute the nearest neighbor distance between a pair of depth images, which form the images (shown as heat maps) in the middle. In this paper, we use $k = 1$. We then apply a classical convolutional neural network to output a score between $(0, 1)$, which is then combined with the status vector to produce the weight of this relative pose according to (10).

cover accurate synchronized transformations $T_i^{(k)}$. We now describe the network design.

Network design. As shown in Figure 3, the key component of our network design is a sub-network $\text{score}_{\theta_0}(S_i, S_j, T_{ij}^{in})$ that takes two scans S_i and S_j and a relative transformation T_{ij}^{in} between them and output a score in $[0, 1]$ that indicates whether this is a good scan alignment or not, i.e., 1 means a good alignment, and 0 means an incorrect alignment.

We design score_{θ_0} as a feed-forward network. Its input consists of two color maps that characterize the alignment patterns between the two input scans. The value of each pixel represents the distance of the corresponding 3D point to the closest points on the other scan under T_{ij}^{in} (See the second column of images in Figure 3). We then concatenate these two color images and feed them into a neural network (we used a modified AlexNet architecture[32]), which outputs the final score.

With this setup, we define the output weight $w_{ij}^{(k+1)}$ as

$$w_{ij}^{(k+1)} := \frac{e^{\theta_1\theta_2}}{e^{\theta_1\theta_2} + (\text{score}_{\theta_0}(S_i, S_j, T_{ij}^{in}) {s_{ij}^{(k)}}^T \theta_3)^{\theta_2}} \quad (10)$$

Note that (10) is conceptually similar to the reweighting scheme $\rho_\sigma(x) = x^2/(\sigma^2 + x^2)$ that is widely used in L^0 minimization (c.f[18]). However, we make elements of the factors and denominators parametric, so as to incorporate status vectors and to capture dataset specific distributions. Moreover, we use exponential functions in (10), since they lead to a loss function that is easier to optimize. With $\theta = (\theta_0, \theta_1, \theta_2, \theta_3)$ we collect all trainable parameters of (10).

4.3. End-to-End Training

Let $\mathcal{D}$ denote a dataset of scan collections with annotated ground-truth poses. Let $k_{\max}$ be the number of recurrent steps (we used four recurrent steps in our experiments) . We define the following loss function for training the weighting module Weight_θ:

$$\min_\theta \sum_{\mathcal{S}\in\mathcal{D}} \sum_{1\le i<j\le|\mathcal{S}|} \Big(\|R_j^{k_{\max}} {R_i^{k_{\max}}}^T - R_j^{gt} {R_i^{gt}}^T\|_{\mathcal{F}}^2 + \lambda \|t_i^{k_{\max}} - t_i^{gt}\|^2 \Big) \quad (11)$$

where we set $\lambda = 10$ in all of our experiments. Note that we compare relative rotations in (11) to factor out the global orientation among the poses. The global shift in translation is already handled by (4).

We perform back-propagation to optimize (11). The technical challenges are to compute the derivatives that pass through the synchronization layer, including 1) the derivatives of $R_j^\star {R_i^\star}^T$ with respect to the elements of L, 2) the derivatives of $t_i^\star$ with respect to the elements of L and $\boldsymbol{b}$, and 3) the derivatives of each status vector with respect to the elements of L and $\boldsymbol{b}$. In the following, we provide explicit expressions for computing these derivatives.

We first present the derivative between the output of rotation synchronization and its input. To make the notation uncluterred, we compute the derivative by treating L as a matrix function. The derivative with respect to w_{ij} can be easily obtained via chain-rule.

Proposition 1. *Let $\boldsymbol{u}_i$ and λ_i be the i-th eigenvector and eigenvalue of L, respectively. Expand the SVD of $U_i = V_i \Sigma_i W_i^T$ as follows:*

$$V_i = (\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \boldsymbol{v}_{i,3}),\ \Sigma_i = \text{diag}(\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3}),$$
$$W_i = (\boldsymbol{w}_{i,1}, \boldsymbol{w}_{i,2}, \boldsymbol{w}_{i,3}).$$

Let $\boldsymbol{e}_j^t \in \mathbb{R}^t$ be the jth canonical basis of $\mathbb{R}^t$. We then have

$$\text{d}(R_j^\star {R_i^\star}^T) = \text{d}R_j \cdot R_i^{\star T} + R_j^\star \cdot \text{d}{R_i}^T,$$

where

$$\text{d}R_i := \sum_{1\le s,t\le 3} \frac{{\boldsymbol{v}_{i,s}}^T \text{d}U_i \boldsymbol{w}_{i,t} - {\boldsymbol{v}_{i,t}}^T \text{d}U_i \boldsymbol{w}_{i,s}}{\sigma_{i,s} + \sigma_{i,t}} \boldsymbol{v}_{i,s} {\boldsymbol{w}_{i,t}}^T,$$

where $\mathrm{d}U_i$ *is defined by* $\forall 1 \leq j \leq 3$,

$$\mathrm{d}U_i \boldsymbol{e}_j^{(3)} = (\boldsymbol{e}_i^{(n)^T} \otimes I_3) \sum_{l=4}^{3n} \frac{\boldsymbol{u}_l \boldsymbol{u}_l^T}{\lambda_j - \lambda_l} \mathrm{d}L \boldsymbol{u}_j.$$

The following proposition specifies the derivative of $\boldsymbol{t}^\star$ with respect to the elements of L and $\boldsymbol{b}$:

Proposition 2. *The derivatives of* $\boldsymbol{t}^\star$ *are given by*

$$\mathrm{d}\boldsymbol{t}^\star = L^+ \mathrm{d}L L^+ + L^+ \mathrm{d}\boldsymbol{b}.$$

Regarding the status vectors, the derivatives of $s_{ij,1}$ with respect to the elements of L are given by Prop. 1; The derivatives of $s_{ij,2}$ and $s_{ij,4}$ with respect to the elements of L are given by Prop. 2. It remains to compute the derivatives of $s_{ij.3}$ with respect to the elements of L, which can be easily obtained via the derivatives of the eigenvalues of L [29], i.e.,

$$\mathrm{d}\lambda_i = \boldsymbol{u}_i^T \mathrm{d}L \boldsymbol{u}_i.$$

5. Experimental Results

This section presents an experimental evaluation of the proposed learning transformation synchronization approach. We begin with describing the experimental setup in Section 5.1. In Section 5.2, we analyze the results of our approach and compare it against baseline approaches. Finally, we present an ablation study in Section 5.3.

5.1. Experimental Setup

Datasets. We consider two datasets in this paper, Redwood [16] and ScanNet [17]:

- **Redwood** contains RGBD sequences of individual objects. We uniformly sample 60 sequences. For each sequence, we sample 30 RGBD images that are 20 frames away from the next one, which cover 600 frames of the original sequence. For experimental evaluation, we use the poses associated with the reconstruction as the ground-truth. We use 35 sequences for training and 25 sequences for testing. Note that the temporal order among the frames in each sequence is discarded in our experiments.

- **ScanNet** contains RGBD sequences, as well as reconstruction, camera pose, for 706 indoor scenes. Each scene contains 2-3 sequences of different trajectories. We randomly sample 100 sequences from ScanNet. We use 70 sequences for training and 30 sequences for testing. Again the temporal order among the frames in each sequence is discarded in our experiments.

More details about the sampled sequences are given in the supplementary material.

Pairwise methods. We consider two state-of-the-art pairwise methods for generating the input to our approach:

- **Super4PCS [34]** applies sampling to find consistent matches of four point pairs.

- **Fast Global Registration (FastGR) [50]** utilizes feature correspondences and applies reweighted nonlinear least squares to extract a set of consistent feature correspondences and fit a rigid pose. We used the Open3D implementation [51].

Baseline approaches. We consider the following baseline approaches that are introduced in the literature for transformation synchronization:

- **Robust Relative Rotation Averaging (RotAvg) [12]** is a scalable algorithm that performs robust rotation averaging of relative rotations. To recover translations, we additionally apply a state-of-the-art translation synchronization approach [26]. We use default setting of its publicly accessible code. [26] is based on our own Python implementation.

- **Geometric Registration (GeoReg) [15]** solve multiway registration via pose graph optimization. We modify the Open3D implementation to take inputs from Super4PCS or FastGR.

- **Transformation Synchronization (TranSyncV2) [9]** is a spectral approach that aims to find a low rank approximation of the null space of the Laplacian matrix. We used the authors' code.

- **Spectral Synchronization in SE(3) (EIGSE3) [7]** is another spectral approach that considers translation and rotation together by working in SE(3). We used the authors' code.

Note that our approach utilizes a weighting module to score the input relative transformations. To make fair comparisons, we use the median nearest-neighbor distances between the overlapping regions (defined as points within distance $0.2m$ from the other point cloud) to filter all input transformations, and select those with median distance below $0.1m$. Note that with smaller threshold the pose graph will be disconnected. We then feed these filtered input transformations to each baseline approach for experimental evaluation.

Evaluation protocol. We employ the evaluation protocols of [11] and [26] for evaluating rotation synchronization and translation synchronization, respectively. Specifically, for rotations, we first solve the best matching global rotation between the ground-truth and the prediction, we then report the statistics and the cumulative distribution function (CDF) of angular deviation $\arccos(\frac{\|\log(R^T R^{gt}\|_{\mathcal{F}}}{\sqrt{2}})$ between

Methods	Redwood												ScanNet											
	Rotation Error						Translation Error (m)						Rotation Error						Translation Error (m)					
	3°	5°	10°	30°	45°	Mean	0.05	0.1	0.25	0.5	0.75	Mean	3°	5°	10°	30°	45°	Mean	0.05	0.1	0.25	0.5	0.75	Mean
FastGR (all)	29.4	40.2	52.0	63.8	70.4	37.4°	22.0	39.6	53.0	60.3	67.0	0.68	9.9	16.8	23.5	31.9	38.4	76.3°	5.5	13.3	22.0	29.0	36.3	1.67
FastGR (good)	33.9	45.2	57.2	67.4	73.2	34.1°	26.7	45.7	58.8	65.9	71.4	0.59	12.4	21.4	29.5	38.6	45.1	68.8°	7.7	17.6	28.2	36.2	43.4	1.43
Super4PCS (all)	6.9	10.1	16.7	39.6	52.3	55.8°	4.2	8.9	18.2	31.0	43.5	1.14	0.5	1.3	4.0	17.4	25.2	98.5°	0.3	1.2	5.3	13.3	21.6	2.11
Super4PCS (good)	10.3	14.9	23.9	48.0	60.0	49.2°	6.4	13.3	26.2	41.2	53.2	0.93	0.8	2.3	6.4	23.0	31.7	90.8°	0.6	2.2	8.9	19.5	29.5	1.80
RotAvg (FastGR)	30.4	42.6	59.4	74.4	82.1	22.4°	23.3	43.2	61.8	72.4	80.7	0.42	6.0	10.4	17.3	36.1	46.1	64.4°	3.7	9.2	19.5	34.0	45.6	1.26
GeoReg (FastGR)	17.8	28.7	47.5	74.2	83.2	27.7°	4.9	18.4	50.2	72.6	81.4	0.93	0.2	0.6	2.8	16.4	27.1	87.2°	0.1	0.7	4.8	16.4	28.4	1.80
RotAvg (Super4PCS)	5.4	8.7	17.4	45.1	59.2	49.6°	3.2	7.4	17.0	32.3	46.3	0.95	0.3	0.8	3.0	15.4	23.3	96.8°	0.2	1.0	5.8	16.5	27.6	1.70
GeoReg (Super4PCS)	2.1	4.1	10.2	33.1	48.3	60.6°	1.1	3.1	10.3	21.5	31.8	1.25	1.9	5.1	13.9	36.6	47.1	72.9°	0.4	2.1	9.8	23.2	34.5	1.82
TranSyncV2 (FastGR)	9.5	17.9	35.8	69.7	80.1	27.5°	1.5	6.2	24.0	48.8	67.5	0.62	0.4	1.5	6.1	29.0	42.2	68.1°	0.2	1.5	11.3	32.0	46.3	1.44
EIGSE3 (FastGR)	36.6	47.2	60.4	74.8	83.3	21.3°	21.5	36.7	57.2	70.4	79.2	0.43	1.5	4.3	12.1	34.5	47.7	68.1°	1.2	4.1	14.7	32.6	46.0	1.29
Our Approach (FastGR)	**67.5**	**77.5**	**85.6**	**91.7**	**94.4**	**6.9°**	20.7	40.0	70.9	**88.6**	**94.0**	0.26	**34.4**	**41.1**	**49.0**	**58.9**	**62.3**	**42.9°**	2.0	7.3	22.3	36.9	48.1	1.16
Our Approach (Super4PCS)	2.3	5.1	13.2	42.5	60.9	46.7°	1.1	4.0	13.8	29.0	42.3	1.02	0.4	1.7	6.8	29.6	43.5	66.9°	0.1	0.8	5.6	16.6	27.0	1.90
Transf. Sync. (FastGR)	27.1	37.7	56.9	74.4	82.4	22.1°	17.4	34.4	55.9	70.4	81.3	0.43	3.2	6.5	14.6	35.8	47.4	63.5°	1.6	5.6	15.5	30.9	43.4	1.31
Input Only (FastGR)	36.7	51.4	68.1	87.7	91.7	13.7°	25.1	**49.3**	73.2	86.4	91.6	0.26	11.7	19.4	30.5	50.7	57.7	51.7°	**5.9**	**15.4**	**30.5**	43.7	52.2	1.03
No Recurrent (FastGR)	37.8	52.8	71.1	87.7	91.7	12.9°	**26.3**	51.1	**77.3**	87.1	92.0	**0.24**	8.6	15.3	26.9	51.4	58.2	49.8°	3.9	11.1	27.3	**43.7**	**53.9**	**1.01**

Figure 4: Benchmark evaluations on Redwood [16] and ScanNet [17]. Quality of absolute poses are evaluated by computing errors to pairwise ground truth poses. Angular distances between rotation matrices are computed via angular $(R_{ij}, R_{ij}^{\star}) = \arccos(\frac{tr(R_{ij}^T R_{ij}^{\star})-1}{2})$. Translation distances are computed by $\|t_{ij} - t_{ij}^{\star}\|$. We collect statistics on percentages of rotation and translation errors that are below a varying threshold. I) The 4th to 7th rows contain evaluations for upstream algorithms. (all) refers to statistics among all pairs where (good) refers to the statistics computed among relative poses with good quality overlap regions. II) For the second part, we report results of all baselines computed from this good set of relative poses, which is consistently better than the results from all relative poses. Since there are two input methods, we report the results of each transformation synchronization approach on both inputs. III) The third parts contain results for ablation study performed only on FastGR[50] inputs. The first row reports state-of-the-art rotation and translation synchronization results, followed by variants of our approach.

a prediction R and its corresponding ground-truth R^{gt}. For translations, we report the statistics and CDF of $\|\boldsymbol{t} - \boldsymbol{t}^{gt}\|$ between each pair of prediction $\boldsymbol{t}$ and its corresponding ground-truth $\boldsymbol{t}^{gt}$. The unit of translation errors are meters (m). The statistics are shown in Figure 4 and the CDF plots are shown in Section B of the supplementary material.

5.2. Analysis of Results

Figure 4 and Figure 5 present quantitative and qualitative results, respectively. Overall, our approach yielded fairly accurate results. On Redwood, the mean errors in rotations/translations of FastGR and our result from FastGR are $34.1°/0.58m$ and $6.9°/0.26m$, respectively. On ScanNet, the mean errors in rotations/translations of FastGR and our result from FastGR are $68.8°/1.43m$ and $42.9°/1.16m$, respectively. Note that in both cases, our approach leads to salient improvements from the input. The final results of our approach on ScanNet are less accurate than those on Redwood. Besides the fact that the quality of the initial relative transformations is lower on ScanNet than that on Redwood, another factor is that depth scans from ScanNet are quite noisy, leading to noisy input (and thus less signals) for the weighting module. Still, the improvements of our approach on ScanNet are salient.

Our approach still requires reasonable initial transformations to begin with. This can be understood from the fact that our approach seeks to perform synchronization by selecting a subset of input relative transformations. Although our approach utilizes learning, its performance shall decrease when the quality of the initial relative transformations drops. An evidence is that our approach only leads to modest performance gains when taking the output of Super4PCS as input.

Comparison with state-of-the-art approaches. Although all the two baseline approaches improve from the input relative transformations, our approach exhibits significant further improvements from all baseline approaches. On Redwood, the mean rotation and translation errors of the top performing method RotAvg from FastGR are $22.4°$ and $0.418m$, respectively. The reductions in mean error of our approach are 69.2% and 39.0% for rotations and translations, respectively, which are significant. The reductions in mean errors of our approach on ScanNet are also noticeable, i.e., 33.3% and 7.4% in rotations and translations, respectively.

Our approach also achieved relative performance gains from baseline approaches when taking the output of Super4PCS as input. In particular, for mean rotation errors, our approach leads to reductions of 5% and 9% on Redwood and ScanNet, respectively.

When comparing rotations and translations, the improvements on mean rotation errors are bigger than those on mean translation errors. One explanation is that there are a lot of planar structures in Redwood and ScanNet. When aligning such planar structures, rotation errors easily lead to a large change in nearest neighbor distances and thus can be

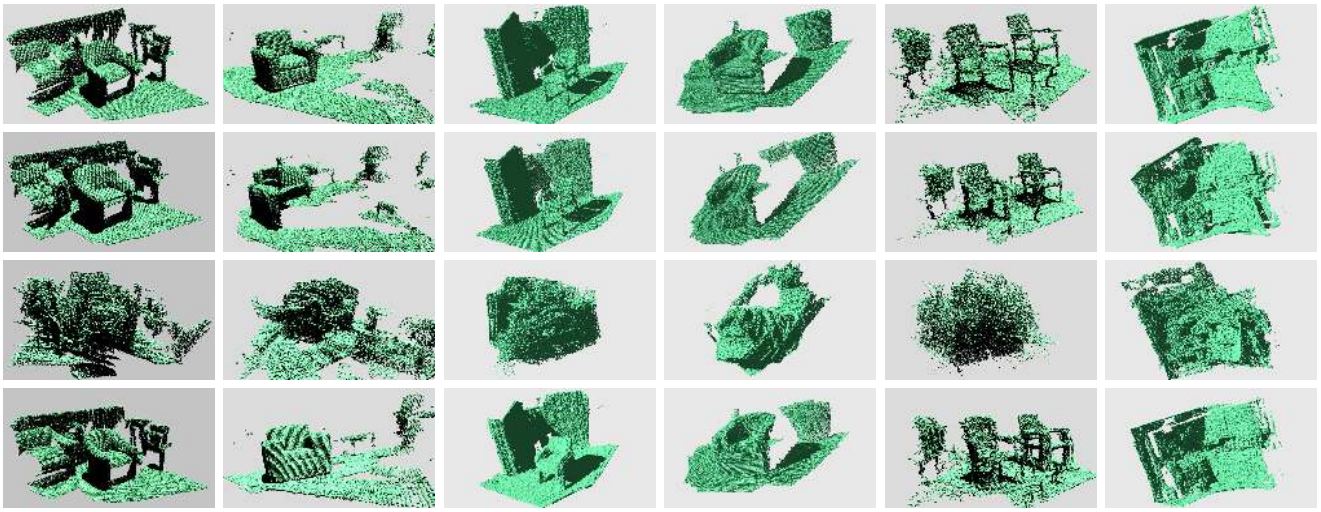

Figure 5: Each column represents the results of one scene. From bottom to top, we show the results of our approach , Rotation Averaging [12]+Translation Sync. [26] (row II), Geometric Registration [15] (row III), and Ground Truth (row IV) (Top). The left four scenes are from Redwood [16] and the right two scenes are from ScanNet [17]

detected by our weighting module. In contrast, translation errors suffer from the gliding effects on planar structures (c.f.[20]). For example, there are rich planar structures that consist of a pair of perpendicular planes, and aligning such planar structures may glide along the common line of these plane pairs. As a result, our weighting module becomes less effective for improving the translation error.

5.3. Ablation Study

In this section, we present two variants of our learning transformation synchronization approach to justify the usefulness of each component of our system. Due to space constraint, we perform ablation study only using FastGR.

Input only. In the first experiment, we simply learn to classify the input maps, and then apply transformation synchronization techniques on the filtered input transformations. In this setting, state-of-the-art transformation synchronization techniques achieves mean rotation/translation errors of $22.1^\circ/0.43m$ and $63.5^\circ/1.25m$ on Redwood and ScanNet, respectively. By applying our learning approach to fixed initial map weights, e.g., we fix θ_0 of the weighting module in (10), our approach reduced the mean errors to $13.7^\circ/0.255m$ and $51.7^\circ/1.031m$ on Redwood and ScanNet, respectively. Although such improvements are noticeable, there are still gaps between this reduced approach and our full approach. This justifies the importance of learning the weighting module together.

No recurrent module. Another reduced approach is to directly combine the weighting module and one synchronization layer. Although this approach can improve from the input transformations. There is still a big gap between this approach and our full approach (See the last row in Figure 4). This shows the importance of using weighting modules to gradually reduce the error while simultaneously make the entire procedure trainable end-to-end.

6. Conclusions

In this paper, we have introduced a supervised transformation synchronization approach. It modifies a reweighted nonlinear least square approach and applies a neural network to automatically determine the input pairwise transformations and the associated weights. We have shown how to train the resulting recurrent neural network end-to-end. Experimental results show that our approach is superior to state-of-the-art transformation synchronization techniques on ScanNet and Redwood for two state-of-the-art pairwise scan matching methods.

There are ample opportunities for future research. So far we have only considered classifying pairwise transformations, it would be interesting to study how to classify high-order matches. Another interesting direction is to install ICP alignment into our recurrent procedure, i.e., we start from the current synchronized poses and perform ICP between pairs of scans to obtain more signals for transformation synchronization. Moreover, instead of maintaining one synchronized pose per scan, we can maintain multiple synchronized poses, which offer more pairwise matches between pairs of scans for evaluation. Finally, we would like to apply our approach to synchronize dense correspondences across multiple images/shapes.

Acknowledgement: The authors wish to thank the support of NSF grants DMS-1546206, DMS-1700234, CHS-1528025, a DoD Vannevar Bush Faculty Fellowship, a Google focused research award, a gift from adobe research, a gift from snap research, a hardware donation from NVIDIA, an Amazon AWS AI Research gift, NSFC (No. 61806176), and Fundamental Research Funds for the Central Universities.

References

[1] Mica Arie-Nachimson, Shahar Z. Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, 3DIMPVT '12, pages 81–88, Washington, DC, USA, 2012. IEEE Computer Society. 4

[2] Federica Arrigoni, Andrea Fusiello, and Beatrice Rossi. Camera motion from group synchronization. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 546–555. IEEE, 2016. 2, 4

[3] Federica Arrigoni, Andrea Fusiello, Beatrice Rossi, and Pasqualina Fragneto. Robust rotation synchronization via low-rank and sparse matrix decomposition. *CoRR*, abs/1505.06079, 2015. 1, 2, 3

[4] Federica Arrigoni, Luca Magri, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 491–498. IEEE, 2014. 2, 4

[5] Federica Arrigoni, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Robust synchronization in SO(3) and SE(3) via low-rank and sparse matrix decomposition. *Computer Vision and Image Understanding*, 174:95–113, 2018. 2

[6] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Global registration of 3d point sets via lrs decomposition. In *European Conference on Computer Vision*, pages 489–504. Springer, 2016. 2

[7] 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, 2, 6

[8] Chandrajit Bajaj, Tingran Gao, Zihang He, Qixing Huang, and Zhenxiao Liang. SMAC: simultaneous mapping and clustering using spectral decompositions. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 334–343, 2018. 2, 14

[9] Florian Bernard, Johan Thunberg, Peter Gemmar, Frank Hertel, Andreas Husch, and Jorge Goncalves. A solution for multi-alignment by transformation synchronisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2169, 2015. 1, 2, 4, 6

[10] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3d SLAM: A survey on rotation estimation and its use in pose graph optimization. In *ICRA*, pages 4597–4604. IEEE, 2015. 1

[11] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *ICCV*, pages 521–528. IEEE Computer Society, 2013. 1, 2, 3, 6

[12] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2018. 1, 6, 8, 12

[13] Yuxin Chen, Leonidas J. Guibas, and Qi-Xing Huang. Near-optimal joint object matching via convex relaxation. In *ICML*, pages 100–108, 2014. 1, 2

[14] Taeg Sang Cho, Shai Avidan, and William T. Freeman. The patch transform. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1489–1501, 2010. 1

[15] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *CVPR*, pages 5556–5565. IEEE Computer Society, 2015. 1, 2, 3, 6, 8, 12

[16] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 2, 6, 7, 8

[17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 1, 2017. 2, 6, 7, 8, 22

[18] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C. Sinan Güntürk. Iteratively re-weighted least squares minimization for sparse recovery. Report, Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, USA, June 2008. 3, 4, 5

[19] Andrea Fusiello, Umberto Castellani, Luca Ronchetti, and Vittorio Murino. Model acquisition by registration of multiple acoustic range views. In *European Conference on Computer Vision*, pages 805–819. Springer, 2002. 2

[20] Natasha Gelfand, Szymon Rusinkiewicz, Leslie Ikemoto, and Marc Levoy. Geometrically stable sampling for the ICP algorithm. In *3DIM*, pages 260–267. IEEE Computer Society, 2003. 8

[21] Venu Madhav Govindu and A Pooja. On averaging multiview relations for 3d scan registration. *IEEE Transactions on Image Processing*, 23(3):1289–1302, 2014. 2

[22] Qixing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. Reassembling fractured objects by geometric matching. *ACM Trans. Graph.*, 25(3):569–578, July 2006. 1, 2

[23] Qixing Huang and Leonidas Guibas. Consistent shape maps via semidefinite programming. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, SGP '13, pages 177–186, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics Association. 1, 2

[24] Qixing Huang, Fan Wang, and Leonidas J. Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Trans. Graph.*, 33(4):36:1–36:11, 2014. 2

[25] Qi-Xing Huang, Guo-Xin Zhang, Lin Gao, Shi-Min Hu, Adrian Butscher, and Leonidas J. Guibas. An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Trans. Graph.*, 31(6):167:1–167:11, 2012. 2

[26] Xiangru Huang, Zhenxiao Liang, Chandrajit Bajaj, and Qixing Huang. Translation synchronization via truncated least squares. In *NIPS*, 2017. 1, 2, 3, 4, 6, 8, 12, 19

[27] Daniel Huber. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, December 2002. 1, 2

[28] Daniel F. Huber and Martial Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21:637–650, 2001. 2

[29] Mohan K. Kadalbajoo and Ankit Gupta. An overview on the eigenvalue computation for matrices. *Neural, Parallel Sci. Comput.*, 19(1-2):129–164, Mar. 2011. 6

[30] Seungryong Kim, Stephen Lin, SANG RYUL JEON, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer

networks for semantic correspondence. In *NIPS*, page to appear, 2018. 2

[31] Vladimir Kim, Wilmot Li, Niloy Mitra, Stephen DiVerdi, and Thomas Funkhouser. Exploring collections of 3d models using fuzzy correspondences. *ACM Trans. Graph.*, 31(4):54:1–54:11, July 2012. 2

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc. 5

[33] Spyridon Leonardos, Xiaowei Zhou, and Kostas Daniilidis. Distributed consistent data association via permutation synchronization. In *ICRA*, pages 2645–2652. IEEE, 2017. 2

[34] Nicolas Mellado, Dror Aiger, and Niloy J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. *Comput. Graph. Forum*, 33(5):205–215, Aug. 2014. 2, 6

[35] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[36] Andy Nguyen, Mirela Ben-Chen, Katarzyna Welnicka, Yinyu Ye, and Leonidas J. Guibas. An optimization approach to improving collections of shape maps. *Comput. Graph. Forum*, 30(5):1481–1491, 2011. 1, 2

[37] Onur Ozyesil and Amit Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015. 1

[38] Deepti Pachauri, Risi Kondor, Gautam Sargur, and Vikas Singh. Permutation diffusion maps (PDM) with application to the image association problem in computer vision. In *NIPS*, pages 541–549, 2014. 1, 2

[39] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. In *NIPS*, pages 1860–1868, 2013. 1, 2

[40] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, pages 292–309, 2018. 2

[41] Gregory C Sharp, Sang W Lee, and David K Wehe. Multiview registration of 3d scenes by minimizing error between coordinate frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1037–1050, 2004. 2

[42] Yanyao Shen, Qixing Huang, Nati Srebro, and Sujay Sanghavi. Normalized spectral map synchronization. In *NIPS*, pages 4925–4933, 2016. 1, 2

[43] Amit Singer and Hau tieng Wu. Vector diffusion maps and the connection laplacian. *Communications in Pure and Applied Mathematics*, 65(8), Aug. 2012. 4

[44] Yifan Sun, Zhenxiao Liang, Xiangru Huang, and Qixing Huang. Joint map and symmetry synchronization. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 257–275, 2018. 2

[45] Chris Sweeney, Torsten Sattler, Tobias Höllerer, Matthew Turk, and Marc Pollefeys. Optimizing the viewing graph for structure-from-motion. In *ICCV*, pages 801–809. IEEE Computer Society, 2015. 1

[46] Andrea Torsello, Emanuele Rodola, and Andrea Albarelli. Multiview registration via graph diffusion of dual quaternions. In *CVPR 2011*, pages 2441–2448. IEEE, 2011. 2

[47] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2:145193, December 2013. 1, 2, 3

[48] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In David J. Fleet, Toms Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV (3)*, volume 8691 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2014. 1

[49] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, pages 1426–1433. IEEE Computer Society, 2010. 2

[50] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 766–782, 2016. 2, 6, 7

[51] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *CoRR*, abs/1801.09847, 2018. 6

[52] Tinghui Zhou, Yong Jae Lee, Stella X. Yu, and Alexei A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, pages 1191–1200. IEEE Computer Society, 2015. 2

[53] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4032–4040, 2015. 2