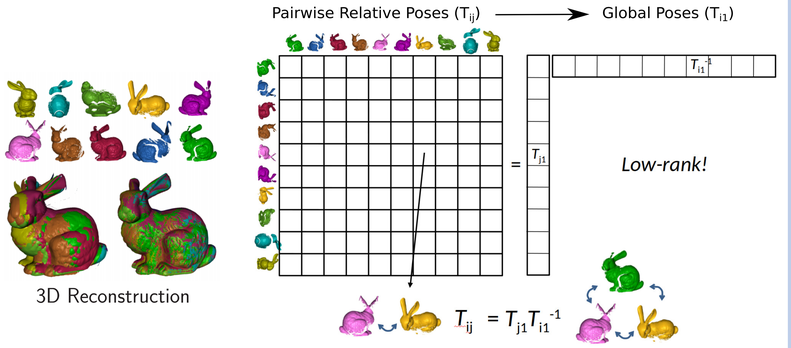
Learning Transformation Synchronization

Xiangru Huang[1], Zhenxiao Liang[1], Xiaowei Zhou[2], Yao Xie[3], Leonidas Guilbas[4] and Qixing Huang[1]
[1] UT Austin [2] Zhejiang University [3] Georgia Tech [4] Stanford

Introduction to Transformation Synchronization



- **Transformation Synchronization** aims at estimating consistent rigid transformations across a collection of images or depth scans.
- Its applications include multi-view structure from motion, geometry reconstruction from depth scans, image editing and simultaneous localization and mapping.

Previous Approaches vs Our Approach

Relative Poses can be noisy.

How to deal with noise?

Existing Assumptions:
1. Noise is random.
2. Only a small fraction of *Relative Poses* are noisy.

Derived Approaches:
1. Least squares, L_1-norm minimization, Pruning with Heuristics.
2. Truncated Least Squares.

Failures:
1. Noise is highly correlated.
2. Majority of Relative Poses can be noisy.

Our Assumption:
Noise patterns are learnable.

Our Approach:
1. Train neural networks to prune bad relative poses.
2. Make the algorithm differentiable.

Our Advantages:
1. Robust to real-world noise.
2. Performance can be improved by by fine-tuning hyperparameters.

Towards learning based approaches

- Given input relative poses $\mathcal{T}^{\text{in}} = \{(R_{ij}^{\text{in}}, t_{ij}^{\text{in}})\}$, a class of iterative reweighting based methods aim to optimize

$$\min_{\{(R_i,t_i)\}} \sum_{1\le i<j\le n} w_{ij}\|R_{ij}^{\text{in}}R_i - R_j\| + w_{ij}\|R_{ij}^{\text{in}}t_i + t_{ij}^{\text{in}} - t_j\|^2 \quad (1)$$

and then at the k-th iteration, update weights w_{ij} by a simple rule, e.g.

$$w_{ij}^k = \begin{cases} 1 & \text{if } \|R_{ij}^{\text{in}}R_i - R_j\| \le \epsilon c^k \\ 0 & \text{otherwise} \end{cases}, \quad c \in (0,1) \quad (2)$$

- Designing a function that distinguish good and bad relative poses is easy only when the noise is random or sparse, but can be much harder in other cases.
- We propose to address this issue by learning the weight update function from data.

Our approach

- Train a binary classifier function $score(scan_i, scan_j, T_{ij}) \in [0,1]$ to distinguish *good* and *bad* relative poses.
- Replace (2) with a soft-thresholding function $reweight(R, R^{\text{in}}, \theta)$ parameterized by θ.

Learning-based Iterative Reweighting Algorithm

Input: Scans $\{S_i\}$, poses $\mathcal{T}^{\text{in}}$ and parameters θ.

0. Initialize the weight by

$$w_{ij}^0 \leftarrow score(S_i, S_j, T_{ij}^{\text{in}})$$

for $k = 1, \ldots, k_{\max}$ **do**

1. Solve a low-rank matrix recovery problem to get absolute rotations $R^k = \{R_i^k | 1 \le i \le n\}$

$$R^k \leftarrow lowrank(w^{k-1}, R^{\text{in}})$$

2. Reweight relative poses

$$w^k \leftarrow reweight(R^k, R^{\text{in}}, \theta)$$

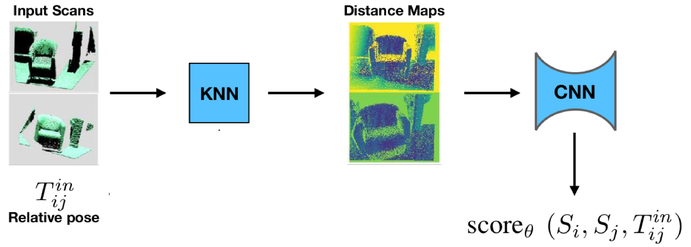
end for

3. Solve Least Squares to obtain translations $\{t_i^{k_{\max}} | 1 \le i \le n\}$.

$$\min_{\{t_i^{k_{\max}}\}} \sum_{1\le i<j\le n} w_{ij}^{k_{\max}}\|R_{ij}^{k_{\max}}t_i^{k_{\max}} + t_{ij}^{\text{in}} - t_j^{k_{\max}}\|^2 \quad (3)$$

return Global poses $\mathcal{T}^\star = \{(R_i^{k_{\max}}, t_i^{k_{\max}}) | 1 \le i \le n\}$

- **Implementing the classifier.** As shown in the figure below, to compute $score(S_i, S_j, T_{ij}^{\text{in}})$, we construct a distance map $M \in \mathbb{R}^{W\times H}$ for each scan. In every entry of the map we fill the point-to-set distance. We then concatenate these two maps and pass it into a Convolutional Neural Network (CNN).



Implementation of classifier

- **Tuning Parameter Vector θ.** When fixing the number of iterations, we can perform gradient descent on parameters θ to optimize pose error.

Experimental Results (Quantitative)

Methods	Redwood		ScanNet	
	Rotation Error	Translation Error (m)	Rotation Error	Translation Error (m)
	Mean	Mean	Mean	Mean
FastGR (all)	37.4°	0.68	76.3°	1.67
FastGR (good)	34.1°	0.59	68.8°	1.43
Super4PCS (all)	55.8°	1.14	98.5°	2.11
Super4PCS (good)	49.2°	0.93	90.8°	1.80
RotAvg (FastGR)	22.4°	0.42	64.4°	1.26
GeoReg (FastGR)	27.7°	0.93	87.2°	1.80
RotAvg (Super4PCS)	49.6°	0.95	96.8°	1.70
GeoReg (Super4PCS)	60.6°	1.25	72.9°	1.82
Our Approach (FastGR)	**6.9°**	**0.26**	**42.9°**	**1.16**
Our Approach (Super4PCS)	46.7°	1.02	66.9°	1.90
Transf. Sync. (FastGR)	22.1°	0.43	63.5°	1.31

Figure: Benchmark evaluations on Redwood (S. Choi, 2016) and ScanNet (A. Dai, 2017).

Experimental Results (Qualitative)

Figure: We show the results of ground truth result (column I), Rotation Averaging (A. Chatterjee, 2017)+Translation Sync. (X. Huang, 2017) (column II), Geometric Registration (S. Choi, 2015) (column III), and Our Approach (column IV). These scenes are from Redwood Chair dataset and ScanNet dataset.