Self-supervised Geometric Perception

Supplementary Material

Heng Yang* MIT LIDS | Wei Dong* CMU RI | Luca Carlone MIT LIDS | Vladlen Koltun Intel Labs

A1. Proof of Proposition 1

Proof. We prove Proposition 1 for Examples 1-2 separately.

Example 1: Relative Pose Estimation. In relative pose estimation, the known geometric model for the i-th measurement pair is $\boldsymbol{R}_i^\circ \in \mathrm{SO}(3)$ and $\boldsymbol{t}_i^\circ \in \mathbb{S}^2$, where $\boldsymbol{R}_i^\circ$ is the relative rotation, and $\boldsymbol{t}_i^\circ$ is the up-to-scale relative translation between two images $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$. Using $(\boldsymbol{R}^\circ, \boldsymbol{t}^\circ)$, we can form the *essential matrix* $\boldsymbol{E}_i^\circ \doteq [\boldsymbol{t}_i^\circ]_\times \boldsymbol{R}_i^\circ$, from which we further compute the *fundamental matrix* $\boldsymbol{F}_i^\circ \doteq (\boldsymbol{K}_i^b)^{-T} \boldsymbol{E}_i^\circ (\boldsymbol{K}_i^a)^{-1}$, where $\boldsymbol{K}_i^a$, $\boldsymbol{K}_i^b$ are the camera intrinsics for the two images $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ [7]. Now we let the residual function $r(\cdot)$ be the algebraic error [7]:

$$r(\boldsymbol{F}_i^\circ, \tilde{\boldsymbol{p}}_{i,k}, \tilde{\boldsymbol{q}}_{i,k}^b) = (\tilde{\boldsymbol{q}}_{i,k}^b)^\mathsf{T} \boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k}, \tag{A1}$$

which should vanish if there is no measurement noise, and $\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}}^b \in \mathbb{R}^3$ denotes the homogeneous coordinates of the keypoint locations. In eq. (A1), $\boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k}$ is called the *epipolar line* (in fact, $\boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k}$ represents the normal vector of the plane formed by the epipolar line and the camera optical center).

Because we have adopted a TLS cost function, *i.e.* $\rho(r) = \min\left\{r^2, \bar{c}^2\right\}$ (and assume $\bar{c}^2$ is small), obviously, the global minimizer of problem (7) is the following:

$$\boldsymbol{q}_{i,k}^b = \mathcal{C}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i, \boldsymbol{p}_i^b, \boldsymbol{b}_i) \in \begin{cases} \text{the epipolar line } \boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k} & \text{if the epipolar line intersects } \boldsymbol{b}_i \\ \boldsymbol{b}_i & \text{otherwise} \end{cases}, \tag{A2}$$

which says that the predicted keypoint $\boldsymbol{q}_{i,k}^b$ should lie precisely on the epipolar line if the epipolar line has a nonempty intersection with the image $\boldsymbol{b}_i$ (so that the residual (A1) is zero and $\rho(r) = 0$), or it can be an arbitrary point on the image otherwise (so that the residual (A1) is nonzero and $\rho(r) = \bar{c}^2$ is very small). In [15], the authors designed another constraint that enforces cycle consistency, *i.e.*, the back-predicted keypoint of the predicted keypoint should be the original keypoint:

$$\mathcal{C}(\boldsymbol{q}_{i,k}^b, \boldsymbol{b}_i, \boldsymbol{p}_i^a, \boldsymbol{a}_i) = \boldsymbol{p}_{i,k}^a. \tag{A3}$$

Combining eq. A2 and (A3), we can reformulate the original feature learning problem (7) as:

$$\text{find} \qquad \mathcal{C}_{\boldsymbol{\theta}} \tag{A4}$$

$$s.t. \quad \mathcal{C} \text{ satisfies (A2) and (A3)}, \tag{A5}$$

which enforces the correspondence function $\mathcal{C}$ (parametrized by $\boldsymbol{\theta} \in \mathbb{R}^{N_\mathcal{C}}$) to map keypoints in $\boldsymbol{a}_i$ to their corresponding epipolar lines (if the epipolar line exists) in $\boldsymbol{b}_i$, and to map the predicted keypoints in $\boldsymbol{b}_i$ back to their original keypoints, which is connected to the cross check criteria mentioned in the main text.

The reformulated problem (A4) is a constrained optimization problem that is not suitable for training neural networks. Therefore, the last step we do is to move the constraints to the cost function and penalize the *violation* of the constraints,

*Equal contribution. Work performed during internship at Intel Labs.

which is commonly referred to as the *Augmented Lagrangian Method* (ALM), or the *penalty method*:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{N_{\mathcal{C}}}} \sum_{i=1}^{M}\sum_{k=1}^{N_{a_i}} \lambda_{\text{epipolar}} \cdot \mathbf{dist}\left(\underbrace{\mathcal{C}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i,\boldsymbol{p}^b_i,\boldsymbol{b}_i)}_{\boldsymbol{q}^b_{i,k}}, \boldsymbol{F}_i^{\circ}\tilde{\boldsymbol{p}}_{i,k}\right)^2 + \lambda_{\text{cycle}} \cdot \mathbf{dist}\left(\mathcal{C}\left(\underbrace{\mathcal{C}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i,\boldsymbol{p}^b_i,\boldsymbol{b}_i)}_{\boldsymbol{q}^b_{i,k}},\boldsymbol{b}_i,\boldsymbol{p}^a_i,\boldsymbol{a}_i\right),\boldsymbol{p}^a_{i,k}\right)^2, \tag{A6}$$

where $\lambda_{\text{epipolar}}, \lambda_{\text{cycle}} > 0$ are constants chosen by the user. Finally, let the correspondence function be the form in (4), we recover the loss function in the CAPS paper [15].[1] Therefore, the CAPS neural network can be seen as a method to solve the feature learning problem (7) by solving its Augmented Lagrangian (A6).

Example 2: Point Cloud Registration. In point cloud registration, the known geometric model for the i-th measurement pairs is the rigid transformation $\boldsymbol{R}_i^{\circ} \in \mathrm{SO}(3)$ and $\boldsymbol{t}_i^{\circ} \in \mathbb{R}^3$ between the two point clouds $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$. Let the residual function $r(\cdot)$ be the Euclidean distance:

$$r(\boldsymbol{R}_i^{\circ},\boldsymbol{t}_i^{\circ},\boldsymbol{p}^a_{i,k},\boldsymbol{q}^b_{i,k}) = \left\|\boldsymbol{q}^b_{i,k} - \boldsymbol{R}_i^{\circ}\boldsymbol{p}^a_{i,k} - \boldsymbol{t}_i^{\circ}\right\|, \tag{A7}$$

which should be zero without measurement noise. Under the TLS cost function $\rho(r) = \min\left\{r^2, \bar{c}^2\right\}$, the global minimizer of problem (7) is

$$\boldsymbol{q}^b_{i,k} = \mathcal{C}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i,\boldsymbol{p}^b_i,\boldsymbol{b}_i) = \begin{cases} \underset{\boldsymbol{p}^b_{i,j}\in\boldsymbol{p}^b_i}{\arg\min}\, r(\boldsymbol{R}_i^{\circ},\boldsymbol{t}_i^{\circ},\boldsymbol{p}^a_{i,k},\boldsymbol{p}^b_{i,j}) & \text{if } \underset{\boldsymbol{p}^b_{i,j}\in\boldsymbol{p}^b_i}{\min}\, r(\boldsymbol{R}_i^{\circ},\boldsymbol{t}_i^{\circ},\boldsymbol{p}^a_{i,k},\boldsymbol{p}^b_{i,j}) < \bar{c} \\ \emptyset & \text{otherwise} \end{cases}, \tag{A8}$$

which states that the correspondence function $\mathcal{C}$ should output the nearest neighbor of $(\boldsymbol{R}_i^{\circ}\boldsymbol{p}^a_{i,k} + \boldsymbol{t}_i^{\circ})$ in $\boldsymbol{p}^b_i$ if the Euclidean distance between the nearest neighbor and $(\boldsymbol{R}_i^{\circ}\boldsymbol{p}^a_{i,k} + \boldsymbol{t}_i^{\circ})$ is close enough to be considered as an inlier, and outputs nothing otherwise (*i.e.*, $\boldsymbol{p}^a_{i,k}$ does not have a corresponding point in $\boldsymbol{p}^b_i$). Therefore, we can reformulate problem (7) as:

$$\text{find} \quad \mathcal{C} \tag{A9}$$
$$s.t. \quad \mathcal{C} \text{ satisfies (A8).} \tag{A10}$$

We then use the fact that $\mathcal{C}$ is a composition of a feature descriptor and nearest neighbor search in the feature space (*cf.* eq. (5) in Example 2), and hence, problem (A9) is further equivalent to finding a descriptor $\mathcal{F}$ such that:

$$\text{find} \quad \mathcal{F} \tag{A11}$$
$$s.t. \quad \mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i),\mathcal{F}(\boldsymbol{q}^b_{i,k},\boldsymbol{b}_i)\right) \le \mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i),\mathcal{F}(\boldsymbol{p}^b_{i,j},\boldsymbol{b}_i)\right), \forall \boldsymbol{p}^b_{i,j} \neq \boldsymbol{q}^b_{i,k}, \tag{A12}$$

which precisely states that the distance in the feature space between $\boldsymbol{p}^a_{i,k}$ and the corresponding keypoint $\boldsymbol{q}^b_{i,k}$ is smaller than the distance between $\boldsymbol{p}^a_{i,k}$ and any other point in $\boldsymbol{p}^b_i$. In fact, we can ask for stronger conditions on the feature descriptor $\mathcal{F}$:

$$\text{find} \quad \mathcal{F} \tag{A13}$$
$$s.t. \quad \mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i),\mathcal{F}(\boldsymbol{q}^b_{i,k},\boldsymbol{b}_i)\right) \le m_p, \tag{A14}$$
$$\mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i),\mathcal{F}(\boldsymbol{p}^b_{i,j},\boldsymbol{b}_i)\right) \ge m_n, \forall \boldsymbol{p}^b_{i,j} \neq \boldsymbol{q}^b_{i,k}, \tag{A15}$$
$$\mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i),\mathcal{F}(\boldsymbol{q}^b_{i,k},\boldsymbol{b}_i)\right) \le m + \mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k},\boldsymbol{a}_i),\mathcal{F}(\boldsymbol{p}^b_{i,j},\boldsymbol{b}_i)\right), \tag{A16}$$

that says: (i) the feature distance between the matched keypoint pair $\boldsymbol{p}^a_{i,k}$ and $\boldsymbol{q}^b_{i,k}$ has to be smaller than a predefined margin $m_p > 0$ (eq. (A14)); (ii) the feature distance between $\boldsymbol{p}^a_{i,k}$ and all the other non-matched keypoints has to be larger than a predefined margin $m_n > m_p$ (eq. (A15)); (iii) the feature distance between non-matched keypoint pairs has to be at least m larger than the feature distance between matched keypoint pairs (eq. (A16)). Obviously, conditions (A14)-(A16) are sufficient (but not necessary) for ensuring condition (A12).

[1] The $\mathbf{dist}\,(\cdot)$ function in (A6) is equivalent to the ℓ_2 norm $\|\cdot\|$. [15] used the $\mathbf{dist}\,(\cdot)$ instead of $\mathbf{dist}\,(\cdot)^2$. This can be easily seen as the Augmented Lagrangian if using the constraint $\sqrt{\mathbf{dist}\,(\cdot)} = 0$, instead of $\mathbf{dist}\,(\cdot) = 0$.

Again, problem (A13) is a constrained optimization that is not suitable for neural network training. Therefore, we develop its Augmented Lagrangian (for the constraints related to the keypoint $\boldsymbol{p}_{i,k}^a$) to be:

$$\begin{aligned}\mathcal{L}(\boldsymbol{p}_{i,k}^a, s_p, s_n, s) = \lambda_p \left(m_p - \mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{q}_{i,k}^b, \boldsymbol{b}_i)\right) - s_p\right)^2 + \\ \sum_{\boldsymbol{p}_{i,j}^b \neq \boldsymbol{q}_{i,k}^b} \lambda_n \left(\mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{p}_{i,j}^b, \boldsymbol{b}_i)\right) - m_n - s_n\right)^2 + \\ \sum_{\boldsymbol{p}_{i,j}^b \neq \boldsymbol{q}_{i,k}^b} \lambda \left(\mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{p}_{i,j}^b, \boldsymbol{b}_i)\right) - \mathbf{dist}\left(\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{q}_{i,k}^b, \boldsymbol{b}_i)\right) - m - s\right)^2, \quad \text{(A17)}\end{aligned}$$

where $s_p, s_n, s \geq 0$ are nonnegative slack variables. In eq. (A17), the first two terms denote the *contrastive loss*, while the last term denotes the *triplet loss*. The ALM [1] solves the following optimization:

$$\min_{\mathcal{F}, s_p \geq 0, s_n \geq 0, s \geq 0} \sum_{i=1}^{M} \sum_{k=1}^{N_{a_i}} \mathcal{L}(\boldsymbol{p}_{i,k}^a, s_p, s_n, s). \quad \text{(A18)}$$

Finally, by enforcing $s_p = s_n = s = 0$, problem (A18) recovers the metric learning problem in the FCGF paper [5]. Therefore, the FCGF neural network can be seen as a method to solve the feature learning problem (7) by solving the Augmented Lagrangian (A18). □

A2. Application of SGP on Object Detection and Pose Estimation

Example A1 (Object Detection and Pose Estimation)**.** *Given a collection of 3D models $\{\boldsymbol{a}_i\}_{i=1}^O$, where each model $\boldsymbol{a}_i \in \mathbb{R}^{3\times N_{a_i}}$ consists of a set of 3D keypoints, let $\boldsymbol{a} \in \mathbb{R}^{3\times N_a}, N_a = \sum_{i=1}^O N_{a_i}$, be the concatenation of all 3D keypoints. In addition, given a corpus of 2D images $\{\boldsymbol{b}_i\}_{i=1}^M$, where each $\boldsymbol{b}_i$ is an RGB image that contains the (partial, occluded) projections of the 3D models plus some background. Object detection and pose estimation seeks to jointly learn a keypoint prediction function $\mathcal{C}$ and estimate the poses of the 3D models $\boldsymbol{x}_i = \{(\boldsymbol{R}_{i,j}, \boldsymbol{t}_{i,j})\}_{j\in\mathcal{S}\subset[O]} \in (\mathrm{SO}(3)\times\mathbb{R}^3)^{|\mathcal{S}|}$, where $\mathcal{S} \subset [O]$ is the subset of 3D models observed by the i-th 2D image ($|\mathcal{S}|$ denotes the cardinality of the set). In particular, following [16], let $\mathcal{C}$ be a combination of UVW mapping and semantic ID masking,* i.e.*, for each pixel in $\boldsymbol{b}_i$, $\mathcal{C}$ predicts which 3D model it belongs to (from 1 to O, and 0 for background), and what is the corresponding 3D coordinates in the specific model, thus deciding which point in $\boldsymbol{a}$ is the corresponding 3D point.*[2]

SGP **for Example A1**. The teacher performs robust absolute pose estimation, *a.k.a. perspective-n-point* (PnP) [7, 9]. A good candidate for the teacher is RANSAC and its variants (*e.g.*, using P3P [6]). The student trains a 2D keypoint predictor under the supervision of camera poses. Recent works such as YOLO6D [14], PVNet [12], and DPOD [16] can all serve as the student network, despite using different methodologies. As for the verifier, similar to Example 1, it can be designed based on the estimated inlier rate by RANSAC. Alternatively, one can project the 3D models onto the 2D image using the estimated absolute poses and compute the overlap ratio (in terms of pixels) between the 2D projection and the estimated semantic ID mask. To initialize SGP, we can train a bootstrap predictor using synthetic datasets, *i.e.*, by rendering synthetic projections of the 3D models under different simulated poses, which is common in [16, 12, 14, 2].

A3. Detailed Experimental Data

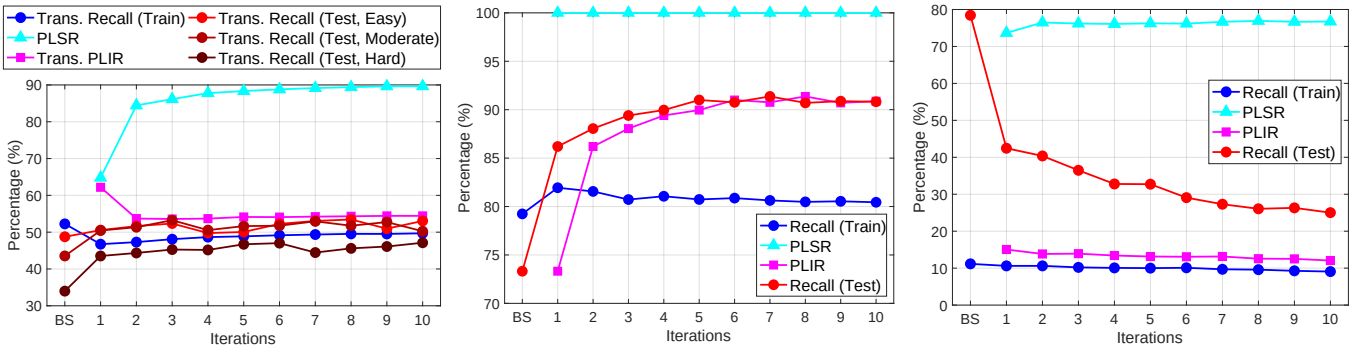
A3.1. Relative Pose Estimation

In Section 5.1, Fig. 2 plots the rotation statistics for running SGP on the MegaDepth [10] dataset for relative pose estimation. Here in Fig. A1(a), we plot the translation statistics. In addition, the full statistics of SGP are tabulated in Table A1. Fig. A2 visualizes 9 qualitative examples of relative pose estimation using S-CAPS^T on the MegaDepth test set.

A3.2. Point Cloud Registration

In Section 5.2, Fig. 4 plots the dynamics of runing SGP on the 3DMatch [17] dataset. Here we provide the full statistics in Table A2.

[2] There are many different ways to establish 2D-3D correspondences, see PVNet [12], YOLO6D [14] and references therein.



(a) SGP translation statistics on MegaDepth. (b) SGP on 3DMatch with exchanged train and test. (c) SGP on 3DMatch with Horn as teacher.

Figure A1. Supplementary statistics. (a) The translation statistics for using SGP on MegaDepth [10] (rotation statistics shown in Fig. 2 in the main text). (b) Dynamics of SGP on 3DMatch [17] with training and test sets exchanged, *i.e.*, we train SGP on the smaller test set (1,623 pairs), but test S-FCGF on the larger training set (9,856 pairs). (c) Dynamics of SGP on 3DMatch by replacing the original RANSAC10K teacher with a non-robust Horn's method [8] as the teacher.

	Statistics (%)	SIFT Bootstrap	SGP trained CAPS (S-CAPS) 1	2	3	4	5	6	7	8	9	10
Train	PLSR	**	64.79	84.44	86.14	87.78	88.34	88.80	89.15	89.41	89.62	89.64
	Rot. PLIR	**	92.50	88.83	88.35	88.41	88.25	88.52	88.50	88.57	88.43	88.48
	Rot. Recall	87.75	79.33	79.69	80.62	80.70	81.20	81.34	81.57	81.57	81.56	81.68
	Trans. PLIR	**	62.20	53.70	53.58	53.68	54.13	54.09	54.22	54.33	54.43	54.43
	Trans. Recall	52.25	46.74	47.30	48.09	48.66	48.87	49.16	49.36	49.54	49.52	49.70
Test Recall	Rot., *Easy*	80.88	85.39	85.49	84.68	85.69	85.79	85.79	86.29	**87.09**	85.49	86.29
	Rot., *Moderate*	58.06	**70.37**	68.27	**70.37**	69.77	69.67	69.27	69.87	68.87	70.07	69.17
	Rot., *Hard*	40.35	48.36	49.38	50.31	49.59	50.10	**51.75**	50.10	50.72	51.23	51.33
	Trans., *Easy*	48.75	50.55	51.65	52.35	49.75	50.05	52.25	53.05	**53.45**	50.95	53.05
	Trans., *Moderate*	43.54	50.45	51.35	**53.25**	50.55	51.65	51.75	52.95	51.75	52.75	50.25
	Trans., *Hard*	33.98	43.53	44.35	45.28	45.17	46.71	47.02	44.46	45.60	46.10	**47.13**

Table A1. Train and test statistics of running SGP on MegaDepth [10]. SGP setting: retrain = False, verifyLabel = True, verifier criteria: number of matches larger than 100 and RANSAC estimated inlier rate larger than 10%. Rotation statistics plotted in Fig. 2 in the main text. Translation statistics plotted in Fig. A1(a).

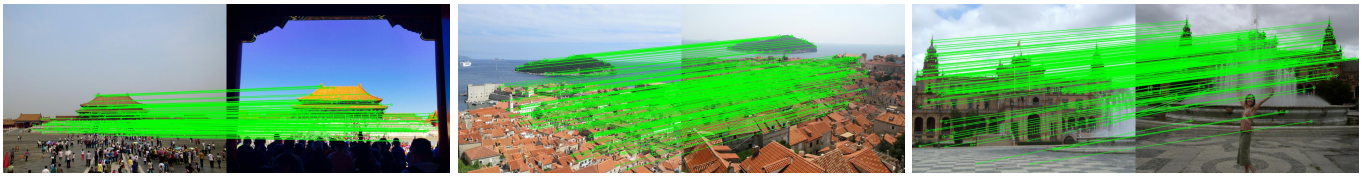
For qualitative results, in Fig. A3 we showcase multiway registration results on various RGB-D datasets [13, 3, 4, 11] in addition to Fig. 5. With S-FCGF, rich loop closures can be detected (in green lines), ensuring high-fidelity camera poses for dense reconstruction. It is worth noting that global registration with trained S-FCGF+RANSAC10K, unlike DGR, can easily run in parallel on a single graphics card due to its inexpensive memory cost. This results in at least 4× speedup comparing to DGR in practice when multi-thread loop closure detection is enabled [18].

A3.3. Ablation Study

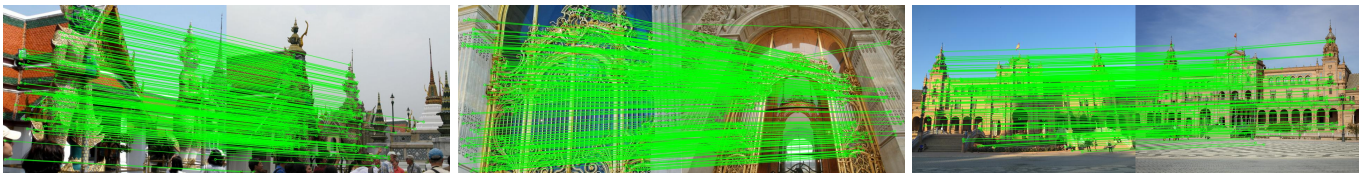
In Section 5.3, Fig. 6 plots the dynamics of running SGP on 3DMatch with two different algorithmic settings: (a) set retrain =True and use retrain instead of finetune; (b) set verifyLabel = False and turn off the verifier. Here we provide the full statistics for (a) and (b) in Table A3 and Table A4, respectively.

Additionally, we show results for two extra ablation experiments on the 3DMatch dataset for point cloud registration.

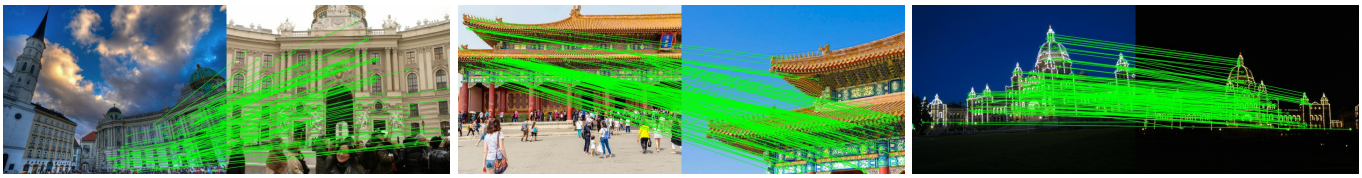
Exchange the training and test sets. Because SGP requires no ground-truth pose labels, there is no fundamental difference between the training and test set, except that the training set (9,856 pairs) is much larger than the test set (1,623 pairs). Therefore, we ask the question: *Can* SGP *learn an equally good feature representation from the much smaller test set?* Our answer is: *it depends on the purpose*. We performed an experiment where we trained SGP on the test set, and tested



(a) *Easy*



(b) *Moderate*



(c) *Hard*

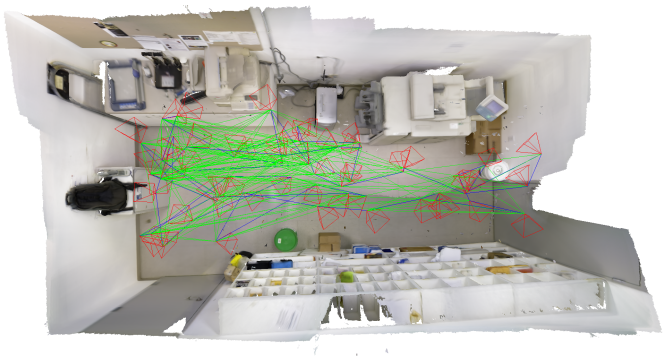
Figure A2. Supplementary qualitative results for relative pose estimation on the MegaDepth dataset [10] using S-CAPS^T.

		FPFH	SGP trained FCGF (S-FCGF)									
Statistics (%)		Bootstrap	1	2	3	4	5	6	7	8	9	10
Train	PLSR	**	69.98	73.91	95.53	95.69	95.74	95.73	95.75	95.73	95.76	95.77
	PLIR	**	92.03	93.42	92.19	92.82	93.02	93.25	93.24	93.43	93.41	93.39
	Recall	82.68	89.14	90.92	91.14	91.43	91.76	91.78	91.95	91.97	91.95	92.05
Test Recall	Kitchen	80.63	98.42	98.02	98.22	98.02	98.22	98.42	98.02	97.83	98.62	98.42
	Home 1	84.62	92.31	93.59	91.03	93.59	92.95	94.23	94.23	94.23	94.23	94.23
	Home 2	69.23	77.88	74.04	75.48	75.00	75.96	73.08	75.96	76.92	73.08	75.00
	Hotel 1	88.05	96.90	97.35	98.23	97.79	98.23	99.12	98.67	98.67	98.23	98.67
	Hotel 2	76.92	87.50	85.58	86.54	90.38	89.42	90.38	90.38	89.42	89.42	89.42
	Hotel 3	88.89	85.19	83.33	83.33	79.63	81.48	79.63	85.19	79.63	77.78	79.63
	Study	71.23	85.27	86.30	87.67	86.99	85.96	86.99	88.01	86.99	86.30	87.33
	MIT	70.13	79.22	79.22	80.52	77.92	77.92	77.92	80.52	76.62	79.22	76.62
	Overall	78.44	90.57	90.14	90.63	90.57	90.57	90.70	**91.37**	90.82	90.45	90.82

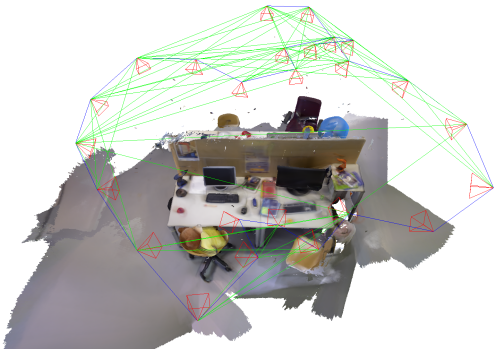
Table A2. Train and test statistics of running SGP on 3DMatch [17]. SGP setting: retrain = False, verifyLabel = True, verifier overlap ratio threshold η: $\eta = 30\%$ for iterations $\tau = 1, 2$, $\eta = 10\%$ for iterations $\tau = 3, \ldots, 10$. Statistics plotted in Fig. 4 in the main text.

the learned S-FCGF representation on the much larger training set. For SGP we used retrain = False and verifyLabel = False. Fig. A1(b) plots the dynamics and Table A5 provides the full statistics. Two observations can be made: (i) Exchanging the training and test set has almost no effect on the recall of S-FCGF on the test set (*cf.* Table A5 vs Table A2-A4). This means that, if one only cares about the performance of the learned representation on the test set, then running SGP directly on the target test set is sufficient. (ii) Although exchanging the training and test set does not hurt the recall on the test set, it indeed decreases the recall on the training set by more than 10%. This suggests that a small training set has the shortcoming of overfitting and the learned representation fails to generalize to a larger dataset. Therefore, if one cares generalization of the learned representation, then a larger training set is still preferred. Nevertheless, this ablation study demonstrates the power of the alternating minimization nature of SGP, that is, SGP is able to find a sufficiently good local minimum.

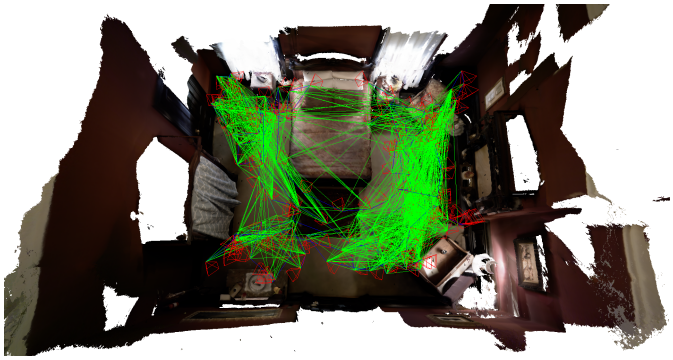
Use a non-robust solver as the teacher. All the experiments so far showed successes of the teacher-student loop, and the robustness of the SGP algorithm to imperfections of both the student and the teacher (noisy geometric pseudo-labels).



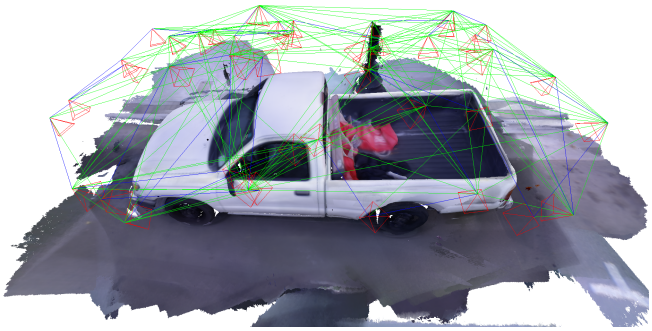
(a) *copyroom* from Stanford RGBD [3].



(b) *long_office* from TUM RGBD [13].



(c) *bedroom* from Indoor LIDAR RGBD [11].



(d) *truck* from Redwood Objects [4].

Figure A3. Supplementary qualitative results for 3D registration. Multi-way reconstruction using S-FCGF+RANSAC10K as the global registration method succeeds on various unseen RGB-D datasets. Blue lines: odometry. Green lines: loop closures.

	Statistics (%)	FPFH Bootstrap	SGP trained FCGF (S-FCGF) 1	2	3	4	5	6	7	8	9	10
Train	PLSR	**	68.48	95.68	95.61	95.61	95.69	95.64	95.60	95.61	95.67	95.65
	PLIR	**	90.86	91.16	92.27	92.40	92.29	92.47	92.52	92.61	92.95	92.44
	Recall	79.24	89.53	90.60	90.69	90.68	90.79	90.77	90.88	91.20	90.72	90.84
Test Recall	Kitchen	**	97.23	97.63	98.22	97.83	98.42	97.83	97.83	97.23	98.42	98.22
	Home 1	**	91.67	93.59	94.23	95.51	94.87	93.59	95.51	95.51	91.03	93.59
	Home 2	**	73.56	71.63	76.92	73.56	75.00	74.04	72.60	76.44	75.00	75.00
	Hotel 1	**	96.90	96.90	96.90	96.46	96.90	96.46	96.90	98.23	97.35	96.90
	Hotel 2	**	85.58	89.42	92.31	88.46	87.50	90.38	88.46	88.46	86.54	91.35
	Hotel 3	**	85.19	88.89	83.33	81.48	83.33	83.33	83.33	85.19	85.19	83.33
	Study	**	82.88	84.59	86.64	88.36	88.70	87.67	87.67	86.30	87.33	86.64
	MIT	**	85.71	83.12	79.22	79.22	83.12	80.52	83.12	77.92	77.92	84.42
	Overall	**	89.34	89.96	91.07	90.57	**91.19**	90.57	90.63	90.70	90.39	90.94

Table A3. Train and test statistics of running SGP on 3DMatch [17]. SGP setting: retrain = True, verifyLabel = True, verifier overlap ratio threshold η: $\eta = 10\%$ for all iterations $\tau = 1, \ldots, 10$. Statistics plotted in Fig. 6(a) in the main text.

However, we ask another question: *Can we, intentionally, make* SGP *fail?* Our answer is: yes if we try badly. We performed an experiment running SGP on 3DMatch, this time replacing RANSAC10K with the non-robust Horn's method [8]. We remark that Horn's method is a subroutine of RANSAC and in practice nobody would use Horn's method alone in the presence of outlier correspondences. Nevertheless, for the purpose of ablation study, we adopted this pessimistic choice. Again, for SGP we used retrain = False, verifyLabel = True with a constant overlap ratio threshold $\eta = 10\%$. Fig. A1(c) shows the dynamics. We

	Statistics (%)	FPFH Bootstrap	SGP trained FCGF (S-FCGF) 1	2	3	4	5	6	7	8	9	10
Train	PLSR	**	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	PLIR	**	79.24	88.82	90.86	91.25	91.63	91.59	91.93	92.12	91.89	91.97
	Recall	79.24	88.82	90.86	91.25	91.63	91.59	91.93	92.12	91.89	91.97	92.05
Test Recall	Kitchen	**	97.43	98.22	98.62	97.83	98.62	98.81	98.22	98.62	98.22	98.22
	Home 1	**	92.31	94.23	91.67	94.23	94.23	92.95	93.59	93.59	94.87	92.95
	Home 2	**	74.04	75.00	72.12	77.40	74.04	74.04	73.56	74.04	73.56	73.08
	Hotel 1	**	95.58	98.23	97.35	97.79	99.12	98.23	98.67	96.90	97.79	97.35
	Hotel 2	**	90.38	93.27	88.46	90.38	88.46	87.50	86.54	88.46	88.46	89.42
	Hotel 3	**	88.89	85.19	83.33	87.04	85.19	85.19	81.48	85.19	83.33	81.48
	Study	**	84.59	87.33	87.67	86.64	88.01	88.01	87.67	88.70	88.70	87.67
	MIT	**	76.62	83.12	77.92	84.42	79.22	80.52	80.52	84.42	83.12	83.12
	Overall	**	89.65	**91.44**	90.26	91.37	91.19	91.00	90.63	91.19	91.13	90.63

Table A4. Train and test statistics of running SGP on 3DMatch [17]. SGP setting: retrain = False, verifyLabel = False. Statistics plotted in Fig. 6(b) in the main text.

	Statistics (%)	FPFH Bootstrap	SGP trained FCGF (S-FCGF) 1	2	3	4	5	6	7	8	9	10
	PLSR	**	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	PLIR	**	73.32	86.20	88.05	89.40	89.96	91.00	90.76	91.37	90.70	90.88
	Recall	73.32	86.20	88.05	89.40	89.96	91.00	90.76	**91.37**	90.70	90.88	90.82
Train	Kitchen	**	94.66	96.84	98.42	98.81	99.21	99.60	99.41	99.01	99.21	99.21
	Home 1	**	91.03	89.74	93.59	95.51	95.51	94.87	94.87	95.51	96.15	95.51
	Home 2	**	70.67	70.67	71.63	69.23	71.15	70.19	74.04	72.60	72.12	72.60
	Hotel 1	**	94.69	96.02	97.35	98.67	98.67	99.12	99.12	99.12	99.12	99.12
	Hotel 2	**	77.88	79.81	77.88	80.77	84.62	83.65	86.54	83.65	84.62	83.65
	Hotel 3	**	83.33	85.19	81.48	85.19	88.89	87.04	85.19	83.33	84.19	85.19
	Study	**	79.79	84.93	87.33	86.64	88.36	87.33	87.67	87.33	87.67	86.99
	MIT	**	75.32	75.32	75.32	79.22	79.22	80.52	80.52	77.92	76.62	79.22
Test on train set		79.24	**81.94**	81.56	80.72	81.06	80.73	80.87	80.63	80.48	80.54	80.44

Table A5. Train and test statistics of running SGP on 3DMatch [17] with **training and test sets exchanged**, *i.e.*, we train SGP on the smaller test set ($1,623$ pairs), but test S-FCGF on the larger training set ($9,856$ pairs). SGP setting: retrain = False, verifyLabel = False. Statistics plotted in Fig. A1(b). We see SGP demonstrates overfitting while training on the smaller test set: S-FCGF achieves equally good (91.37%) recall on the test set, but only achieves below 82% recall on the training set (while in Tables A2-A4 S-FCGF has over 92% recall on the training set). Statistics plotted in Fig. A1(b).

see that the PLIR is always below 20%, meaning that 8 out of 10 geometric labels passed to FCGF training are wrong. In this case, the learned S-FCGF representation keeps getting worse, as shown by the decreasing recalls on both the training and test set. Note that for testing, we actually used RANSAC10K as the registration solver to be consistent with other experiments we performed on 3DMatch. However, even with RANSAC10K, the test recall drops to below 30%. Therefore, this ablation study shows the necessity of a robust teacher for SGP to work. Fortunately, we have plenty of robust solvers, as discussed in the main text. So we think this is a strength of SGP, rather than a weakness.

References

[1] Dimitri Bertsekas. *Nonlinear programming*. Athena Scientific, 1999. 3

[2] Bo Chen, Jiewei Cao, Alvaro Parra, and Tat-Jun Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 3

[3] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, 2015. 4, 6

[4] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 4, 6

[5] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Intl. Conf. on Computer Vision (ICCV)*, pages 8958–8966, 2019. 3

[6] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(8):930–943, 2003. 3

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 1, 3

[8] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer.*, 4(4):629–642, Apr 1987. 4, 6

[9] Laurent Kneip, Hongdong Li, and Yongduek Seo. UPnP: An optimal o(n) solution to the absolute pose problem with universal applicability. In *European Conf. on Computer Vision (ECCV)*, pages 127–142. Springer, 2014. 3

[10] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2050, 2018. 3, 4, 5

[11] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *ICCV*, 2017. 4, 6

[12] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise voting network for 6dof pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4561–4570, 2019. 3

[13] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 4, 6

[14] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 292–301, 2018. 3

[15] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *European Conf. on Computer Vision (ECCV)*, 2020. 1, 2

[16] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. DPOD: 6d pose object detector and refiner. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1941–1950, 2019. 3

[17] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and T Funkhouser. 3dmatch: Learning the matching of local 3d geometry in range scans. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 3, 4, 5, 6, 7

[18] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 4