Appendix: Solving the Blind Perspective-n-Point Problem End-To-End With Robust Differentiable Geometric Optimization

Dylan Campbell, Liu Liu, and Stephen Gould

Australian National University, Australian Centre for Robotic Vision

1 Analytic Derivatives

Sinkhorn Layer: The lower-level objective function [2] for the Sinkhorn layer, given a cost matrix $\mathbf{M} \in \mathbb{R}_+^{m \times n}$, is

$$f(\mathbf{M}, \mathbf{P}) = \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbf{M}_{ij} \mathbf{P}_{ij} + \mu \mathbf{P}_{ij} (\log \mathbf{P}_{ij} - 1)) \tag{1}$$

subject to $\mathbf{P} \in U(\mathbf{r}, \mathbf{c})$, where the transport polytope

$$U(\mathbf{r}, \mathbf{c}) = \{\mathbf{P} \in \mathbb{R}_+^{m \times n} \mid \mathbf{P}\mathbf{1}^n = \mathbf{r},\, \mathbf{P}^\mathsf{T}\mathbf{1}^m = \mathbf{c}\} \tag{2}$$

is defined for the prior probability vectors $\mathbf{r} \in \mathbb{R}_+^m$ and $\mathbf{c} \in \mathbb{R}_+^n$ with $\sum \mathbf{r} = 1$ and $\sum \mathbf{c} = 1$.

We can write the optimization problem as

$$\begin{aligned} \mathbf{p}^\star = \underset{\mathbf{P}}{\arg\min} \quad & f(\mathbf{m}, \mathbf{p}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{p} = \mathbf{d} \\ & \mathbf{p} \geqslant 0 \end{aligned} \tag{3}$$

where $\mathbf{p}$ is the vectorized (flattened) form of $\mathbf{P}$, $\mathbf{m}$ is the vectorized form of $\mathbf{M}$, and

$$\mathbf{A} = \widetilde{\mathbf{A}}_{-i} \tag{4}$$

$$\mathbf{d} = \widetilde{\mathbf{d}}_{-i} \tag{5}$$

$$\widetilde{\mathbf{A}} = \left(\begin{array}{c|c|c} \mathbf{e}_1, \ldots, \mathbf{e}_1 & \ldots & \mathbf{e}_m, \ldots, \mathbf{e}_m \\ \hline \mathbf{I}_n & \ldots & \mathbf{I}_n \end{array}\right) \tag{6}$$

$$\widetilde{\mathbf{d}} = \begin{pmatrix} \mathbf{r} \\ \mathbf{c} \end{pmatrix} \tag{7}$$

where the subscript $-i$ denotes removal of the i^{th} (any) row, and $\widetilde{\mathbf{A}} \in \mathbb{R}^{(m+n) \times (mn)}$ is a $2 \times m$ block matrix of standard basis vectors $\mathbf{e}_i \in \mathbb{R}^m$ and identity matrices $\mathbf{I}_n \in \mathbb{R}^{n \times n}$.

For reference, we replicate the relevant lemma from the main paper.

Lemma 1. *Consider a function* $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ *and let* $\mathbf{A} \in \mathbb{R}^{p \times m}$ *and* $\mathbf{d} \in \mathbb{R}^p$ *with rank*$(\mathbf{A}) = p$ *define a set of* p *under-constrained linear equations* $\mathbf{Au} = \mathbf{d}$. *Also let* $\mathbf{y}(\mathbf{x}) \in \arg\min_{\mathbf{u}} f(\mathbf{x}, \mathbf{u})$ *subject to* $\mathbf{Au} = \mathbf{d}$. *Assume that* $\mathbf{y}(\mathbf{x})$ *exists and that* $f(\mathbf{x}, \mathbf{u})$ *is second-order differentiable in the neighborhood of* $\mathbf{u} = \mathbf{y}(\mathbf{x})$. *Set* $\mathbf{H} = \mathrm{D}^2_{YY} f(\mathbf{x}, \mathbf{y})$ *and* $\mathbf{B} = \mathrm{D}^2_{XY} f(\mathbf{x}, \mathbf{y})$. *Then*

$$\mathrm{D}\mathbf{y}(\mathbf{x}) = \left(\mathbf{H}^{-1}\mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^\mathsf{T})^{-1}\mathbf{A}\mathbf{H}^{-1} - \mathbf{H}^{-1}\right)\mathbf{B}. \tag{8}$$

Therefore, given optimal $\mathbf{p}^\star$, corresponding to $\mathbf{y}$ in Lemma 1, we can compute the derivative $\mathrm{D}\mathbf{p}^\star(\mathbf{m})$ using (8). We do not need to consider the positivity constraints, because optimizing the objective function always generates a non-negative solution $\mathbf{P}^\star$. The matrix $\mathbf{A}$ has already been defined, and the matrices $\mathbf{B}$ and $\mathbf{H}$ can be derived as

$$\mathbf{B} = \mathbf{I}_{mn} \in \mathbb{R}^{mn \times mn} \tag{9}$$

$$\mathbf{H} = \mathrm{diag}\left\{\frac{\mu}{\mathbf{p}_{ij}}\right\} \in \mathbb{R}^{mn \times mn}. \tag{10}$$

The Hessian $\mathbf{H}$ is trivially invertible.

While the derivative can easily be computed using these matrices and Cholesky inversion to compute $(\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^\mathsf{T})^{-1}$, the memory and computation efficiency of this approach is extremely inefficient. For example, the memory requirements of this naïve approach are greater than $O(m^2n^2)$.

We reduce the memory requirements to $O(mn)$ by instead decomposing the relevant matrices into block form, and using the sparse structure of the problem to only store the necessary elements. For example, we store the inverse Hessian as a vector in $\mathbb{R}^{mn}$ and never form the matrices $\mathbf{A}$ or $\mathbf{B}$ explicitly. In particular, we exploit the block structure of $\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^\mathsf{T}$ to avoid computing the costly inverse of this $(m+n) \times (m+n)$ matrix directly, instead using blockwise inversion to significantly reduce the amount of computation. Finally, we do not compute the derivative $\mathrm{D}\mathbf{p}^\star(\mathbf{m}) \in \mathbb{R}^{mn \times mn}$ explicitly. Instead, we build the vector–Jacobian product $\mathrm{D}_P L(\mathbf{p}^\star)\mathrm{D}\mathbf{p}^\star(\mathbf{m})$ from left to right, never storing a matrix larger than $m \times n$.

2 Additional Results

Weaker (pose-only) supervision: Here we explore the types of supervision that can be used with our model: pose-only and pose + correspondences. Since both datasets provide ground-truth 2D–3D correspondences, we use them as an additional source of supervision in the main paper. However, weaker pose-only supervision can also be used, which expands the applicability of our method, since ground-truth correspondences are difficult to obtain. Weak supervision via the backprojection of 3D points into the image provides noisy correspondences, which does not handle occlusions. Occlusions are rare in the sparse MegaDepth dataset ($> 95\%$ of pose-estimated correspondences match the ground-truth labels) and so weak supervision is unlikely to change the performance. However,

Table 1. Results on test set of the synthetic ModelNet40 [9] dataset. We report quartiles for rotation error (°), translation error and reprojection error (°), and the mean runtime (in seconds). Results using the RANSAC estimate are denoted with R and those using weaker pose-only supervision are denoted with $L_{\mathrm{c}}^{\star}$. †Algorithms were run for a maximum of 30s.

	Rotation Error			Translation Error			Reprojection Error			Time
Method	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	$\bar{x}$
SoftPOSIT [3]	16.1	21.8	28.0	0.33	0.49	0.72	2.82	3.98	5.21	27†
RANSAC [4]	90.8	139	165	0.43	1.15	3.08	4.22	5.87	8.06	30†
GOSMA [1]	10.1	22.1	52.0	0.25	0.46	0.75	1.04	1.62	3.11	30†
Ours $L_{\mathrm{c}}^{\star}$	5.91	11.35	17.99	0.19	0.33	0.56	0.45	0.76	1.21	0.12
Ours L_{c}	6.08	11.34	18.33	0.34	0.52	0.81	0.56	0.86	1.31	0.12
Ours $L_{\mathrm{c}}^{\star}R$	5.19	11.03	19.06	**0.04**	0.09	0.19	0.35	0.67	1.19	0.12
Ours $L_{\mathrm{c}}R$	5.49	11.67	20.04	**0.04**	0.09	0.20	0.37	0.70	1.25	0.12
Ours $L_{\mathrm{c}}^{\star}R$ + LM	4.80	10.33	18.38	**0.04**	0.10	0.21	0.33	0.64	1.16	0.12
Ours $L_{\mathrm{c}}R$ + LM	5.07	11.08	19.45	**0.04**	0.10	0.22	0.35	0.67	1.23	0.12
Ours $L_{\mathrm{c}}^{\star}L_{\mathrm{p}}$	5.87	11.07	17.38	0.05	0.10	0.21	0.39	0.68	1.13	0.13
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	4.88	9.66	16.01	**0.04**	**0.08**	**0.15**	0.36	0.61	1.03	0.12
Ours $L_{\mathrm{c}}^{\star}L_{\mathrm{p}}R$	4.61	10.18	17.69	**0.04**	0.09	0.19	0.33	0.63	1.12	0.13
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	3.33	8.09	15.82	**0.04**	**0.08**	0.16	0.28	0.52	1.01	0.12
Ours $L_{\mathrm{c}}^{\star}L_{\mathrm{p}}R$ + LM	4.26	9.69	17.29	0.05	0.10	0.21	0.32	0.60	1.10	0.13
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	**3.09**	**7.60**	**15.28**	**0.04**	0.09	0.17	**0.27**	**0.50**	**0.98**	0.12
Ground-truth	0	0	0	0	0	0	0.18	0.18	0.18	–

there are more occlusions in the dense ModelNet40 dataset. Nonetheless, we only get a small reduction of 2° (rotation) and 0.01 (translation) on the median statistics with weak supervision and still outperform all SOTA methods significantly. We report the full results in Table 1, where pose-only supervision is denoted $L_{\mathrm{c}}^{\star}$ (since the correspondence loss does not use ground-truth correspondence labels).

Pose refinement: Here we explore the utility of additional post-processing for improving the pose estimation accuracy. Specifically, we refine the pose estimate for inlier 2D–3D point pairs using the Levenberg–Marquadt (LM) algorithm [6]. Inliers are determined using a 1° threshold from the RANSAC [4] pose estimate. The results for the ModelNet40 [9] dataset are presented in Table 1 and for the MegaDepth [7] dataset in Table 2. The refinement process improves the rotation and reprojection errors but makes little difference to the translation error. This is not unexpected, because LM optimizes the reprojection error directly, and small translation perturbations do not affect this measure strongly.

Convergence of globally-optimal algorithms: On the large datasets we are using, it is not feasible to run the globally-optimal algorithms until convergence, since this would require months of computation. Even with the 30s limit, evalu-

Table 2. Results on test set of the real-world MegaDepth [7] dataset. We report quartiles for rotation error (°), translation error and reprojection error (°), and the mean runtime (in seconds). †Algorithms were run for a maximum of 30s.

	Rotation Error			Translation Error			Reprojection Error			Time
Method	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	$\bar{x}$
SoftPOSIT [3]	1.81	21.4	165	0.24	1.53	6.10	0.92	7.85	24.1	18†
RANSAC [4]	66.6	122	155	6.80	15.2	28.2	4.45	8.77	13.3	30†
GOSMA [1]	8.69	86.8	145	1.07	5.67	9.34	1.30	13.7	37.1	30†
Ours L_c	1.91	4.47	11.39	0.52	1.05	2.34	0.54	1.12	2.81	0.23
Ours $L_c L_p$	1.32	3.31	8.84	0.21	0.46	1.08	0.21	0.53	1.64	0.22
Ours $L_c R$	0.44	1.55	7.70	0.05	0.18	0.80	**0.06**	0.16	1.27	0.23
Ours $L_c R$ + LM	0.34	1.31	7.27	0.05	0.18	0.82	**0.06**	0.13	1.16	0.23
Ours $L_c L_p R$	0.34	1.00	4.88	**0.04**	**0.12**	**0.53**	**0.06**	0.12	0.74	0.22
Ours $L_c L_p R$ + LM	**0.26**	**0.88**	**4.68**	**0.04**	0.13	0.59	**0.06**	**0.11**	**0.64**	0.22
Ground-truth	0	0	0	0	0	0	0.02	0.02	0.03	–

ation takes 5 days per method. However, we would still like to quantify this 30s limit, to see whether there is a runtime–accuracy trade-off that can be obtained. To do so, we ran GOSMA on 5% of the ModelNet40 test set for 10min each. This only improved the median results by 2.7° (rotation) and 0.04 (translation), still far from the results we obtain in this paper, which indicates that the 30s limit is not unreasonable.

Alternative feature matching strategy: At inference time, we can apply alternative feature matching strategies instead of Sinkhorn normalization. Here we explore the use of the Hungarian algorithm [5] to solve the assignment problem exactly, enforcing strict one-to-one correspondences, as a drop-in replacement for the Sinkhorn layer. Indeed, the Sinkhorn algorithm can be viewed as solving a relaxation of the assignment problem. Note however that the (non-relaxed) assignment problem has non-differentiable constraints and so the gradient cannot be back-propagated through the Hungarian algorithm during end-to-end learning. Differentiability is not required during inference, however. We present the results for the ModelNet40 and MegaDepth datasets in Tables 3 and 4. Using the Hungarian algorithm improves the results, particularly for the MegaDepth dataset, at the expense of a slight time penalty due to the higher computational complexity.

Outliers: Here we explore the effect of two types of outliers on our pre-trained network. First, we add ωm and ωn random outliers to the 2D and 3D point-sets respectively, for an outlier fraction ω. The outliers are drawn uniformly from the bounding box enclosing the point-sets, and represent incorrect detections in the image and 3D model. The results are shown in Table 5 and Figure 1. They indicate that the method has some inherent robustness to outliers, with

Table 3. Hungarian feature matching. Results on test set of the synthetic ModelNet40 [9] dataset, with the Hungarian algorithm replacing the declarative Sinkhorn layer (denoted "+H"). We report quartiles for rotation error (°), translation error and reprojection error (°), and the mean runtime (in seconds).

	Rotation Error			Translation Error			Reprojection Error			Time
Method	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	$\bar{x}$
Ours L_c	6.08	11.34	18.33	0.34	0.52	0.81	0.56	0.86	1.31	**0.12**
Ours L_c+H	**5.90**	**11.21**	**18.28**	**0.08**	**0.20**	**0.43**	**0.39**	**0.70**	**1.18**	0.17
Ours L_cL_p	4.88	9.66	16.01	0.04	0.08	0.15	0.36	0.61	1.03	**0.12**
Ours L_cL_p+H	**4.74**	**9.55**	**15.88**	**0.03**	**0.07**	**0.13**	**0.32**	**0.58**	**1.00**	0.15
Ours L_cR	5.49	11.67	20.04	**0.04**	**0.09**	0.20	0.37	**0.70**	**1.25**	**0.12**
Ours L_cR+H	**5.38**	**11.46**	**19.73**	**0.04**	**0.09**	**0.19**	**0.35**	0.72	**1.25**	0.17
Ours L_cL_pR	**3.33**	**8.09**	**15.82**	0.04	**0.08**	**0.16**	**0.28**	**0.52**	**1.01**	**0.12**
Ours L_cL_pR+H	3.79	8.79	16.29	**0.03**	**0.08**	**0.16**	0.29	0.55	1.05	0.15

Table 4. Hungarian feature matching. Results on test set of the real-world MegaDepth [7] dataset, with the Hungarian algorithm replacing the declarative Sinkhorn layer (denoted "+H"). We report quartiles for rotation error (°), translation error and reprojection error (°), and the mean runtime (in seconds).

	Rotation Error			Translation Error			Reprojection Error			Time
Method	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	$\bar{x}$
Ours L_c	1.91	4.47	11.39	0.52	1.05	2.34	0.54	1.12	2.81	**0.23**
Ours L_c+H	**1.12**	**3.66**	**10.90**	**0.15**	**0.49**	**1.43**	**0.15**	**0.58**	**2.06**	0.45
Ours L_cL_p	1.32	3.31	8.84	0.21	0.46	1.08	0.21	0.53	1.64	**0.22**
Ours L_cL_p+H	**0.82**	**2.91**	**8.57**	**0.09**	**0.32**	**0.94**	**0.09**	**0.40**	**1.48**	0.46
Ours L_cR	0.44	1.55	7.70	0.05	0.18	0.80	0.06	0.16	1.27	**0.23**
Ours L_cR+H	**0.24**	**0.78**	**3.67**	**0.03**	**0.09**	**0.42**	**0.05**	**0.10**	**0.54**	0.45
Ours L_cL_pR	0.34	1.00	4.88	0.04	0.12	0.53	0.06	0.12	0.74	**0.22**
Ours L_cL_pR+H	**0.19**	**0.53**	**2.20**	**0.02**	**0.06**	**0.25**	**0.04**	**0.08**	**0.33**	0.46

satisfactory performance up to $\omega = 0.5$, with a median error less than 10°. Second, we add ωn structured outliers to the 3D point-sets, selected at random from the set of 3D model points that do not correspond to any 2D point, due to occlusion, blurring, or the restricted field-of-view of the camera. The results are shown in Table 6 and Figure 2. The observed level of robustness is similar to the case of random outliers. It is very likely that including random or structured outliers during training would improve the robustness of the method at inference time. However, we wished to explore the inherent robustness of the method, since

Table 5. Random 2D and 3D outliers. Results on test set of the real-world MegaDepth [7] dataset, with random outlier points added to the 2D and 3D point-sets. We report quartiles for rotation error (°), translation error and reprojection error (°), for a given outlier fraction ω.

		Rotation Error			Translation Error			Reprojection Error		
Method	ω	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0	1.32	3.31	8.84	0.21	0.46	1.08	0.21	0.53	1.64
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0	0.34	1.00	4.88	0.04	0.12	0.53	0.06	0.12	0.74
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0	0.26	0.88	4.68	0.04	0.13	0.59	0.06	0.11	0.64
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.1	4.28	7.80	14.55	0.72	1.43	2.84	0.70	1.48	2.79
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.1	0.90	3.15	10.50	0.12	0.37	1.16	0.10	0.35	1.80
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.1	0.70	2.76	10.07	0.10	0.36	1.17	0.09	0.29	1.71
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.2	5.58	9.60	17.03	0.94	1.83	3.70	0.91	1.96	3.54
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.2	1.82	5.27	13.65	0.22	0.64	1.58	0.18	0.71	2.53
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.2	1.47	4.86	13.34	0.20	0.62	1.58	0.15	0.62	2.47
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.3	6.23	10.60	18.51	1.07	2.11	4.23	1.07	2.29	4.08
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.3	2.86	7.04	16.28	0.36	0.89	2.04	0.30	1.08	3.15
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.3	2.50	6.64	15.97	0.33	0.87	2.04	0.23	1.00	3.11
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.4	6.71	11.41	19.77	1.18	2.31	4.57	1.17	2.55	4.54
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.4	3.96	8.77	18.17	0.51	1.13	2.37	0.44	1.41	3.75
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.4	3.57	8.46	17.92	0.48	1.11	2.38	0.37	1.34	3.72
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.5	7.02	11.94	20.59	1.26	2.43	4.82	1.29	2.77	4.89
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.5	4.73	10.03	19.86	0.61	1.32	2.69	0.57	1.69	4.26
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.5	4.43	9.69	19.54	0.59	1.30	2.70	0.49	1.64	4.22

one of the benefits of a declarative approach is the ability to include RANSAC within the data processing pipeline.

LiDAR dataset: We also trained and tested the model on the Data61/2D3D dataset [8], a dataset of paired 3D LiDAR point-sets and 2D panoramic images with 10 distinct outdoor scenes, of which 3 were reserved as the test set (IDs 2, 6 and 10). Each panoramic image was converted to a set of regular images with a 72° field-of-view to increase the challenge of the dataset. We used a semantic segmentation to extract, in 2D and 3D, static objects that do not lie on the ground plane, for example, buildings and trees, and then randomly downsample to 1000 points and pixels. Note that the 3D model has no associated visual information, and so a blind PnP approach is appropriate for this data. The results are given in Table 7 and show that the model works well on unstructured laser rangefinder data.

Qualitative results: Additional qualitative results for the MegaDepth dataset [9] are provided in Figures 3, 4, 5, and 6, including several failure cases. This

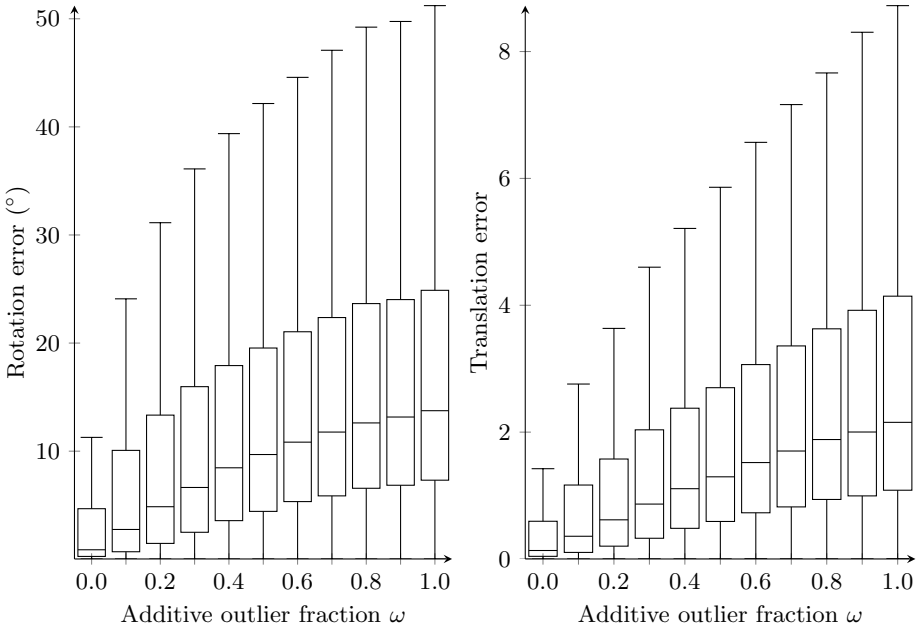

Fig. 1. Random 2D and 3D outliers. Rotation (left) and translation (right) error with respect to the test-time additive outlier fraction ω on the MegaDepth dataset, using our best model with LM refinement (Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM). We do not visualize the outlier errors, defined as any point $> Q3 + 1.5(Q3 - Q1)$, to ensure a good scale for visualization.

covers every scene in the test set. Recall that SoftPOSIT [3] is initialized close to the ground-truth camera pose, and so possesses privileged information and is not directly comparable.

Two videos of sample point-sets from the MegaDepth dataset [9], with the camera frusta found by GOSMA (red), our method (blue), and the ground-truth (black), are included in the supplementary material folder. The ground-truth and our camera frusta overlap completely.

Experimental setup details: The GOSMA algorithm [1] requires a translation domain to be provided, since the search space is otherwise unbounded. We select a search space that encompasses all reasonable camera positions, including the ground-truth camera location, without being too large. To do so, we compute the coordinates of an axis-aligned bounding box that includes all 3D points except outliers (points with coordinates below the 2.5^{th} percentile or above the 97.5^{th} percentile); we extend the bounding box to include the ground-truth camera position; and we expand the resulting translation domain by 10%.

Table 6. Structured 3D outliers. Results on test set of the real-world MegaDepth [7] dataset, with structured outlier points added to the 3D point-sets. We report quartiles for rotation error (°), translation error and reprojection error (°), for a given outlier fraction ω.

		Rotation Error			Translation Error			Reprojection Error		
Method	ω	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0	1.32	3.31	8.84	0.21	0.46	1.08	0.21	0.53	1.64
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0	0.34	1.00	4.88	0.04	0.12	0.53	0.06	0.12	0.74
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0	0.26	0.88	4.68	0.04	0.13	0.60	0.06	0.11	0.64
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.1	3.53	7.17	15.83	0.55	1.11	2.62	0.75	1.33	2.53
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.1	1.09	3.78	11.88	0.14	0.45	1.30	0.15	0.47	1.73
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.1	0.84	3.25	11.39	0.11	0.41	1.31	0.13	0.38	1.66
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.2	4.09	8.07	17.13	0.63	1.29	3.09	0.92	1.62	2.95
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.2	1.73	5.43	15.11	0.22	0.68	1.75	0.23	0.78	2.37
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.2	1.34	4.91	14.75	0.19	0.64	1.76	0.18	0.68	2.31
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.3	4.62	8.83	18.55	0.68	1.44	3.48	1.07	1.85	3.30
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.3	2.59	7.28	18.29	0.32	0.90	2.29	0.35	1.13	3.02
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.3	2.13	6.78	18.20	0.28	0.87	2.33	0.28	1.05	2.96
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.4	4.91	9.54	19.43	0.75	1.56	3.75	1.18	2.05	3.58
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.4	3.40	8.91	20.73	0.43	1.15	2.73	0.50	1.44	3.58
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.4	2.95	8.62	20.77	0.38	1.12	2.76	0.41	1.36	3.53
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	0.5	5.23	9.99	19.90	0.81	1.65	3.91	1.29	2.24	3.81
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.5	4.31	10.29	23.65	0.54	1.38	3.14	0.66	1.76	4.09
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	0.5	3.87	10.01	23.75	0.51	1.35	3.19	0.57	1.68	4.05

Table 7. Results on test set of the LiDAR Data61/2D3D [8] dataset. We report quartiles for rotation error (°), translation error (metres) and reprojection error (°), and the mean runtime (in seconds).

	Rotation Error			Translation Error			Reprojection Error			Time
Method	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	$\bar{x}$
Ours L_{c}	2.14	4.91	10.37	2.12	3.99	7.99	1.17	2.05	3.70	0.14
Ours $L_{\mathrm{c}}L_{\mathrm{p}}$	1.96	4.28	9.50	1.64	3.00	6.52	0.94	1.67	3.15	0.14
Ours $L_{\mathrm{c}}R$	0.36	1.18	7.16	0.19	0.88	4.95	**0.11**	0.34	2.59	0.14
Ours $L_{\mathrm{c}}R$ + LM	0.31	0.92	6.71	0.18	0.74	4.58	**0.11**	0.28	2.48	0.14
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$	0.35	1.02	5.89	0.17	0.73	4.20	**0.11**	0.28	2.10	0.14
Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM	**0.30**	**0.80**	**5.41**	**0.16**	**0.63**	**3.82**	**0.11**	**0.24**	**1.95**	0.14

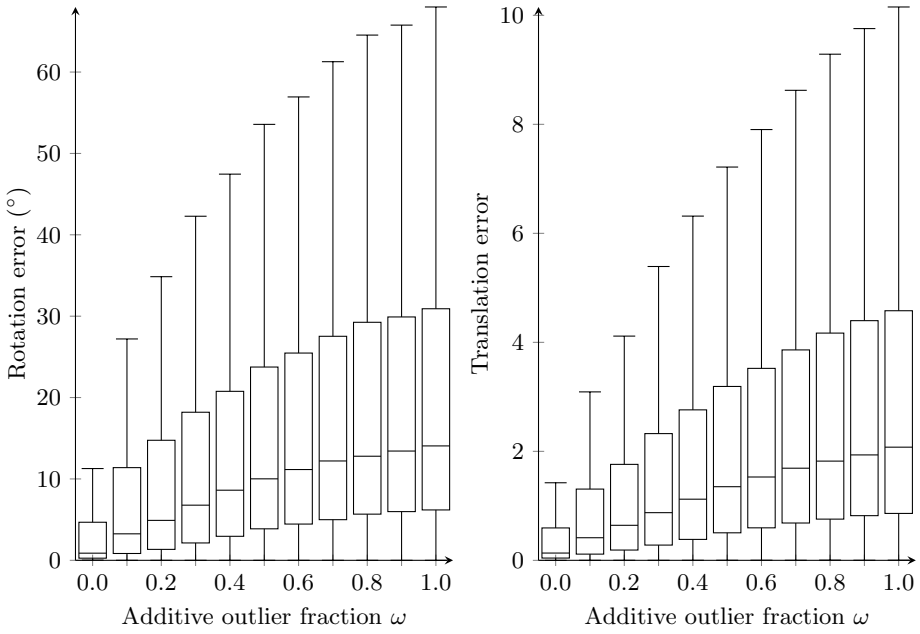

Fig. 2. Structured 3D outliers. Rotation (left) and translation (right) error with respect to the test-time additive outlier fraction ω on the MegaDepth dataset, using our best model with LM refinement (Ours $L_{\mathrm{c}}L_{\mathrm{p}}R$ + LM). We do not visualize the outlier errors, defined as any point $> Q3 + 1.5(Q3 - Q1)$, to ensure a good scale for visualization.

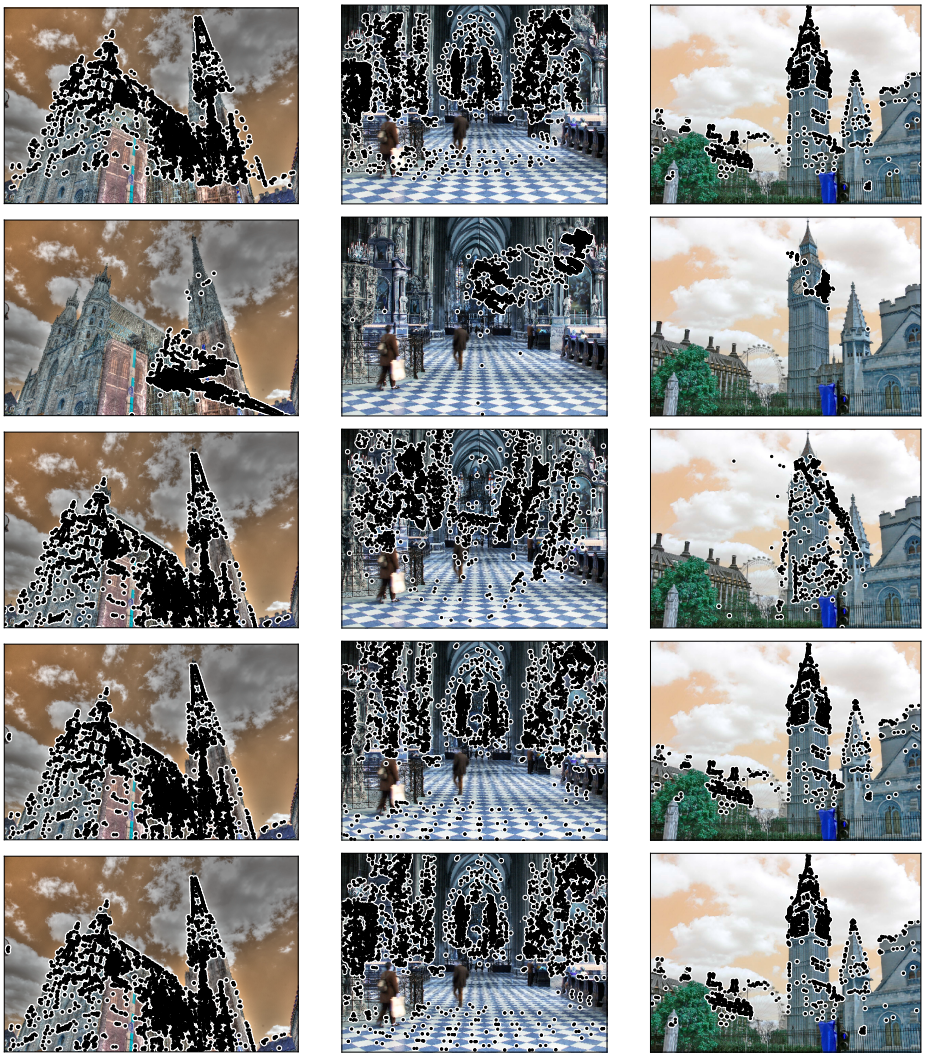

Fig. 3. Qualitative results for the MegaDepth dataset. The 3D point-sets are projected onto the image plane using the estimated camera poses found using (from top to bottom) SoftPOSIT [3], RANSAC [4], GOSMA [1], our method, and the ground-truth.

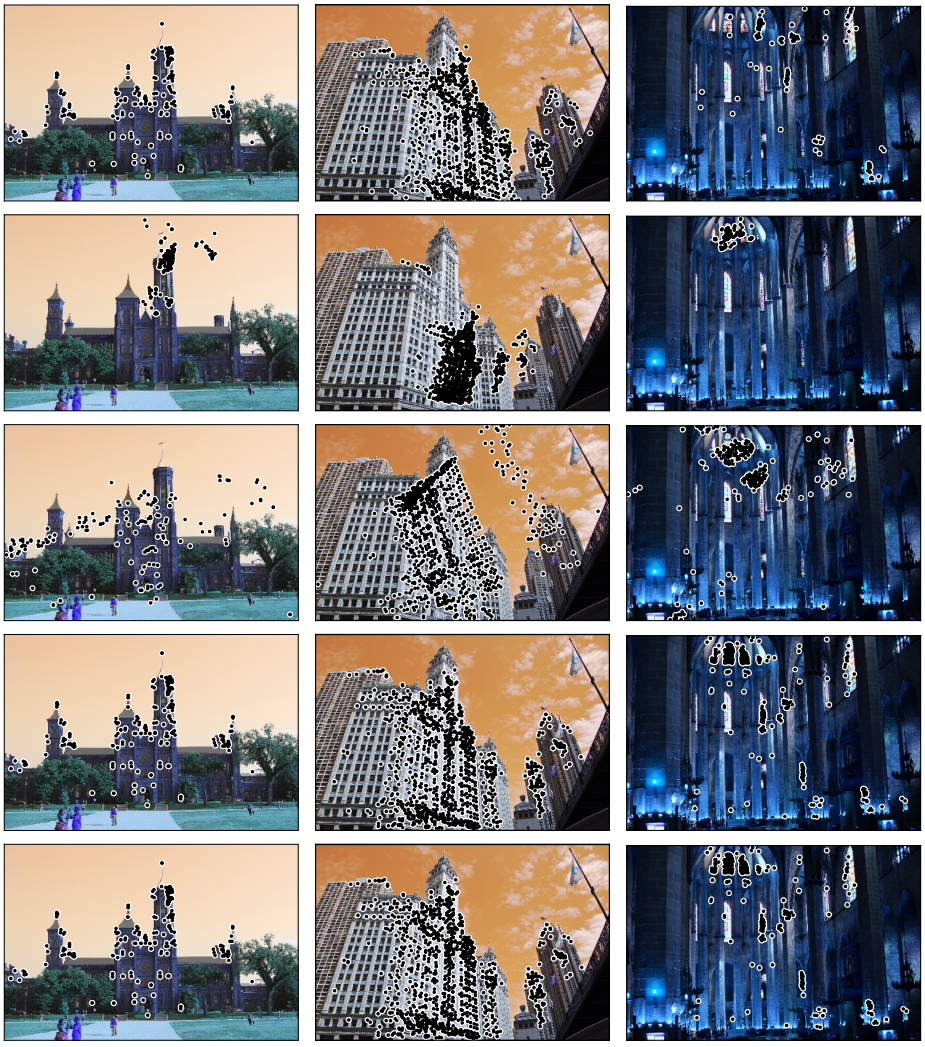

Fig. 4. Qualitative results for the MegaDepth dataset. The 3D point-sets are projected onto the image plane using the estimated camera poses found using (from top to bottom) SoftPOSIT [3], RANSAC [4], GOSMA [1], our method, and the ground-truth.

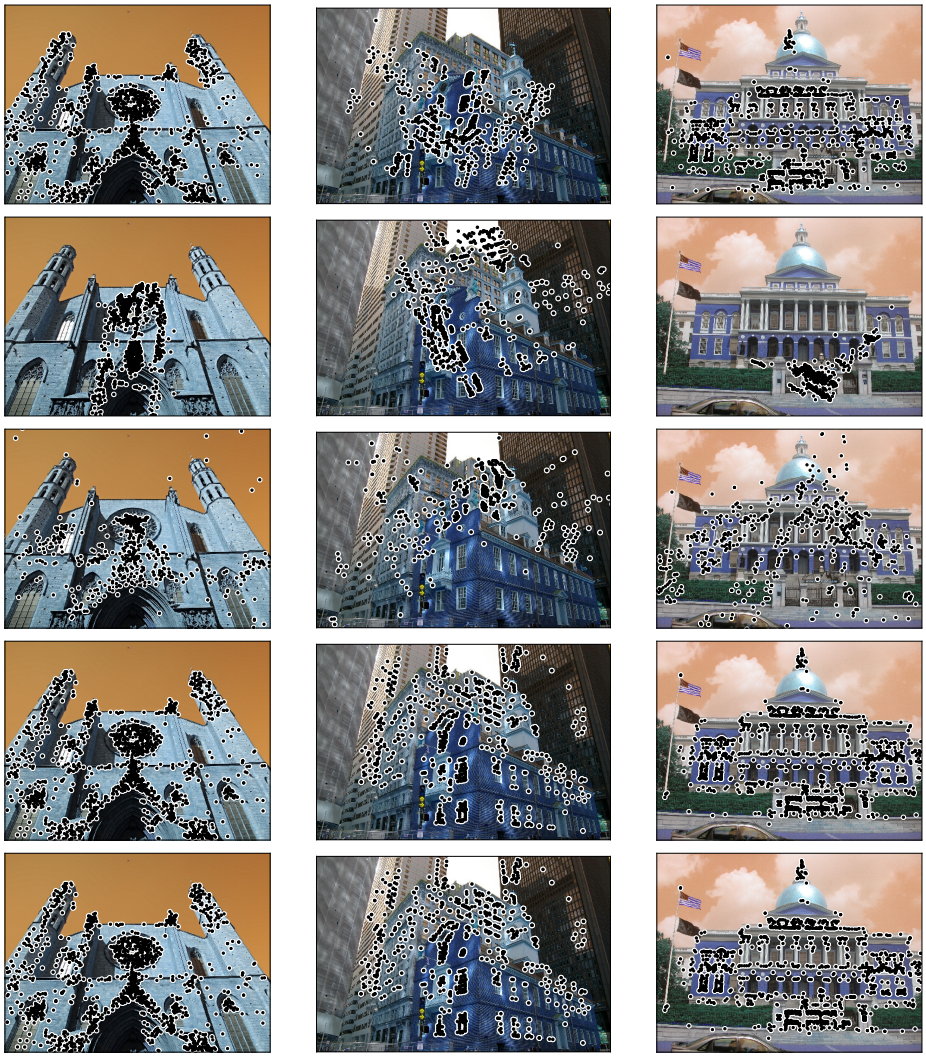

Fig. 5. Qualitative results for the MegaDepth dataset. The 3D point-sets are projected onto the image plane using the estimated camera poses found using (from top to bottom) SoftPOSIT [3], RANSAC [4], GOSMA [1], our method, and the ground-truth.

Fig. 6. Qualitative results for the MegaDepth dataset. The 3D point-sets are projected onto the image plane using the estimated camera poses found using (from top to bottom) SoftPOSIT [3], RANSAC [4], GOSMA [1], our method, and the ground-truth.

References

1. Campbell, D., Petersson, L., Kneip, L., Li, H., Gould, S.: The alignment of the spheres: Globally-optimal spherical mixture alignment for camera pose estimation. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11796–11806 (2019)
2. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 2292–2300 (2013)
3. David, P., Dementhon, D., Duraiswami, R., Samet, H.: SoftPOSIT: simultaneous pose and correspondence determination. International Journal of Computer Vision (IJCV) **59**(3), 259–284 (2004)
4. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
5. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
6. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Quarterly of Applied Mathematics **2**(2), 164–168 (1944)
7. Li, Z., Snavely, N.: MegaDepth: Learning single-view depth prediction from internet photos. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2041–2050 (2018)
8. Namin, S.T., Najafi, M., Salzmann, M., Petersson, L.: A multi-modal graphical model for scene analysis. In: Proceedings of the 2015 Winter Conference on Applications Computer Vision. pp. 1006–1013. IEEE (Jan 2015)
9. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A deep representation for volumetric shapes. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1912–1920 (2015)