Appendix

In the following pages, we present additional quantitative results, qualitative results and experimental details about the Neural Reprojection Error.

A. Additional Experiments

A.1. NRE-based pose estimator *vs*. Feature metric Pose Refinement

We compare our novel NRE-based pose estimator against Feature-Metric Pose Refinement (FPR) methods. As explained in Section 7.2, FPR methods seek to minimize Eq. 12. As such, FPR benefits from dense information contained in query feature maps, but requires to choose a robust loss function and tune its hyperparameters.

To complement our RE-based *vs*. NRE-based pose estimators study presented in Tab. 1, we propose to reuse S2DNet [19] features to perform FPR, initialized from our best RE pose estimator (MAGSAC++ [5]). To merge information from all three feature extraction levels from S2DNet [19], we try upsampling and concatenating descriptors, as well as a coarse-to-fine alternative in which we iteratively refine predictions from the previous (coarser) level.

We report pose estimation errors in Tab. 4 for FPR and NRE estimators. We show results using the Huber [21] robust loss as well as the Barron [4] loss. We find that NRE performs consistently better while eliminating the need for choosing a robust loss.

A.2. Experiments on Aachen Night [35]

So far, we evaluated the performances of our NRE-based pose estimator on MegaDepth [25]. Here, we run a similar study on the Aachen Night [35, 37] dataset. This challenging outdoor dataset consists of $4,328$ sparsely sampled daytime database images, and 98 nighttime query images. To have a fair comparison between NRE-based and RE-based pose estimators, we pair each query image with an oracle nearest-neighbor database image and use all of its visible 3D points to predict the query pose. Similar to the MegaDepth study, we report results for RE-based, FPR-based and NRE-based pose estimators, using S2DNet features in Tab. 5. For FPR-based pose estimators we pick the best configuration from 4.

As in the MegaDepth experiment, our NRE-based pose estimator consistently provides significant improvement over other pose estimators. We also compare the performance coupling the NRE-based pose estimator with NRE features trained on the same training set as S2DNet [19]. We report in Tab. 6 the pose estimation errors. We again find that using NRE features brings an additional leap in performance.

A.3. Experiments on InLoc [46]

To evaluate the generalization capabilities in an indoor scenario, we run the same experiment on the InLoc [46] dataset. This dataset consists of 329 query images, for $9,972$ database images. Unlike Aachen Night, we have access to dense aligned depth maps for all database images. To provide a fair comparison, we also pair each query image with an oracle nearest-neighbor database image and use SuperPoint [16] detections (lifted to 3D using the depth maps) in the database images as inputs. Results are reported in Tab. 5.

We find that our NRE-based pose estimator provides consistent improvements at the coarsest threshold, and overall competitive performance on the medium and fine ones. The fact the relative improvement brought by our NRE-based pose estimator is not as significant as for the other datasets can be attributed to the domain shift with respect to the training images. Nonetheless, despite being trained on outdoor images we find that our NRE features bring additional improvements compared to S2DNet [19] features, as shown in Tab. 6.

B. Qualitative results

In Fig. 6, we show several examples of query images from the MegaDepth [25] validation set with a reprojected 3D point and the corresponding coarse dense loss map computed using our coarse NRE features. It highlights that the dense loss maps keep much more information than RE. As a consequence, as we show in our experiments, our novel NRE-based pose estimator significantly outperforms RE-based pose estimators.

C. Derivation of Equation 9

In this section, we show how Eq. 8 (in the submited wersion of the paper) is obtained.

The robust dense loss map $\mathtt{L}_{\mathtt{Q},n,\sigma}$ can be smoothed using an isotropic Gaussian kernel as follows:

$$
\begin{aligned}
&\breve{\mathtt{L}}_{\mathtt{Q},n,\sigma}\left(\mathbf{p}\right) := \sum_{\mathbf{r}} k_\sigma\left(\|\mathbf{r}\|\right) \mathtt{L}_{\mathtt{Q},n}\left(\mathbf{p}+\mathbf{r}\right) \\
&= \mathtt{L}_{\mathtt{Q},n}\left(\mathbf{out}\right) \sum_{\mathbf{r}} k_\sigma\left(\|\mathbf{r}\|\right) \\
&\quad + \sum_{\mathbf{r}} k_\sigma\left(\|\mathbf{r}\|\right) \left(\mathtt{L}_{\mathtt{Q},n}\left(\mathbf{p}+\mathbf{r}\right) - \mathtt{L}_{\mathtt{Q},n}\left(\mathbf{out}\right)\right) \qquad (13) \\
&= \sum_{\mathbf{r}} k_\sigma\left(\|\mathbf{r}\|\right) \left(\mathtt{L}_{\mathtt{Q},n}\left(\mathbf{p}+\mathbf{r}\right) - \mathtt{L}_{\mathtt{Q},n}\left(\mathbf{out}\right)\right) + \mathrm{cst}_{\mathbf{p}} \qquad (14) \\
&= \sum_{\mathbf{q}\in\Omega_{\mathtt{Q}}} k_\sigma\left(\|\mathbf{q}-\mathbf{p}\|\right) \left(\mathtt{L}_{\mathtt{Q},n}\left(\mathbf{q}\right) - \mathtt{L}_{\mathtt{Q},n}\left(\mathbf{out}\right)\right) + \mathrm{cst}_{\mathbf{p}} \\
& \qquad (15)
\end{aligned}
$$

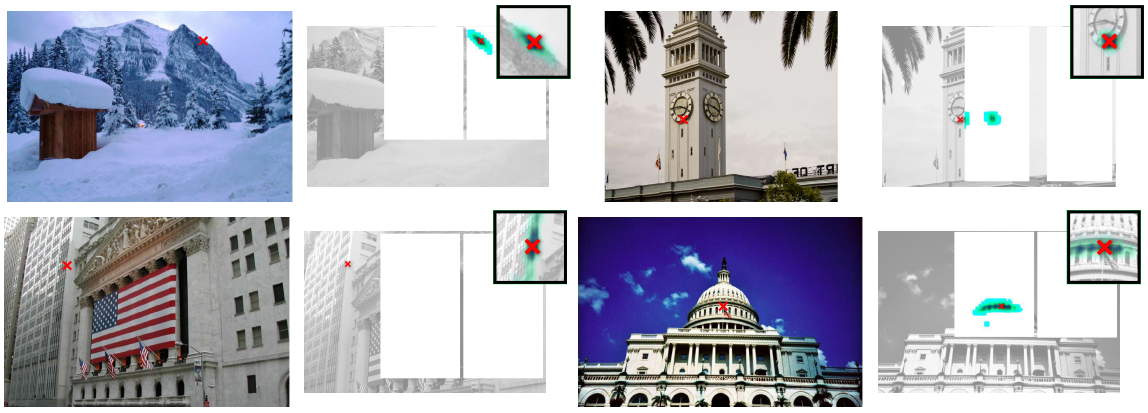

Figure 6. **Qualitative results:** These qualitatives results correspond to additional examples for columns (a) and (b) in Fig.1. It highlights that the dense loss maps keep much more information than RE. As a consequence our novel NRE-based pose estimator significantly outperforms RE-based pose estimators.

Features	Pose estimator	Fusion	ψ	Translation Error			Rotation Error		
				0.25m	1m	5m	2°	5°	10°
S2DNet [19]	RE *MAGSAC++* [5]	N/A	N/A	0.51 (+ 16%)	0.43 (+ 26%)	0.31 (+ 24%)	0.51 (+ 16%)	0.45 (+ 22%)	0.42 (+ 24%)
S2DNet [19]	FPR *Min. Eq. 12*	C2F	Huber [21]	0.70 (+ 59%)	0.65 (+ 91%)	0.52 (+108%)	0.69 (+ 57%)	0.63 (+ 70%)	0.58 (+ 71%)
S2DNet [19]	FPR *Min. Eq. 12*	C2F	Barron [4]	0.55 (+ 25%)	0.44 (+ 29%)	0.30 (+ 20%)	0.55 (+ 25%)	0.48 (+ 30%)	0.43 (+ 26%)
S2DNet [19]	FPR *Min. Eq. 12*	Concat.	Huber [21]	0.49 (+ 11%)	0.42 (+ 24%)	0.30 (+ 20%)	0.48 (+ 9%)	0.44 (+ 19%)	0.42 (+ 24%)
S2DNet [19]	FPR *Min. Eq. 12*	Concat.	Barron [4]	0.49 (+ 11%)	0.42 (+ 24%)	0.30 (+ 20%)	0.48 (+ 9%)	0.44 (+ 19%)	0.42 (+ 24%)
S2DNet [19]	NRE	N/A	N/A	**0.44** (+ 0%)	**0.34** (+ 0%)	**0.25** (+ 0%)	**0.44** (+ 0%)	**0.37** (+ 0%)	**0.34** (+ 0%)

Table 4. **NRE-based pose estimator *vs*. Feature-Metric Pose Refinement:** We evaluate the gain in performance of our novel NRE-based pose estimator against the Feature-Metric Pose Estimation (FPR) variant on the MegaDepth dataset. Here FPR consists in minimizing Eq. 12 using as initialization the camera pose estimate from RE *MAGSAC++* [5]. We find here that minimizing Eq. 12 allows to improve the camera pose estimate from *MAGSAC++*, however our novel NRE again shows superior performance, while requiring no robust kernel selection. The scores between brackets show the relative deterioration w.r.t. to NRE.

$$= \sum_{\mathbf{q}\in\Omega_{\mathtt{Q}}} k_\sigma\left(\|\mathbf{q}-\mathbf{p}\|\right)\left(\mathtt{L}_{\mathtt{Q},n}\left(\mathbf{q}\right) - \ln|\mathring{\Omega}_{\mathtt{Q}}|\right) + \mathrm{cst}_{\mathbf{p}} \quad (16)$$

$$= \sum_{\mathbf{q}\in\Gamma_{\mathtt{Q},n}} k_\sigma\left(\|\mathbf{q}-\mathbf{p}\|\right)\left(\mathtt{L}_{\mathtt{Q},n}\left(\mathbf{q}\right) - \ln|\mathring{\Omega}_{\mathtt{Q}}|\right) + \mathrm{cst}_{\mathbf{p}} \quad (17)$$

where $k_\sigma\left(\|\mathbf{r}\|\right) := \frac{1}{2\pi\sigma^2} e^{-\frac{\|\mathbf{r}\|^2}{2\sigma^2}}$ is an isotropic Gaussian kernel with standard variation σ and $\Gamma_{\mathtt{Q},n}$ is the set of pixel locations whose corresponding values in $\mathtt{L}_{\mathtt{Q},n}$ are lower than $\ln|\mathring{\Omega}_{\mathtt{Q}}|$. Equation 17 leads to the smoothed cost function:

$$\breve{\mathcal{L}}_\sigma\left(\mathtt{R}_{\mathtt{QG}}, \mathbf{t}_{\mathtt{QG}}\right) :=$$
$$\sum_{n=1}^{N}\sum_{\mathbf{q}\in\Gamma_{\mathtt{Q},n}} \text{-}\left(\ln|\mathring{\Omega}_{\mathtt{Q}}| \text{-} \mathtt{L}_{\mathtt{Q},n}\left(\mathbf{q}\right)\right) k_\sigma\left(\left\|\mathbf{q} \text{-} \omega\left(\mathbf{u}_n^{\mathtt{G}}, \mathtt{R}_{\mathtt{QG}}, \mathbf{t}_{\mathtt{QG}}\right)\right\|\right), \quad (18)$$

which is a robust non-linear least squares problem and therefore can be minimized using the IRLS algorithm.

D. Technical details

D.1. Coarse-to-Fine Strategy (Sec. 5.3)

Step 6 of our coarse-to-fine strategy consists in computing local high-resolution loss maps of size 64×64 at the location of the reprojected 3D points using the coarse pose estimate. The idea of that step is to transform the low-resolution loss maps into high-resolution loss maps to obtain a much more accurate pose estimate. The question is: How can we combine a low-resolution robust loss map with a local high-resolution discriminative loss map ? We proceed as follows:

1. A coarse correspondence map $\mathtt{C}_{\text{coarse}}$ is of size $H/16 \times W/16$. Let us recall that by definition $\sum_{\mathbf{p}\in\mathring{\Omega}_{\text{coarse}}} \mathtt{C}_{\text{coarse}}\left(\mathbf{p}\right) = 1$.

2. Compute the local high resolution correspondence map $\mathtt{C}_{\text{fine}}$ of size 64×64 at the location of the reprojected 3D points (using the coarse pose estimate) $\mathbf{q}$:

Features	Pose Estimator	Aachen Night			InLoc-DUC1			InLoc-DUC2		
		0.25m, 2°	0.5m, 5°	5m, 10°	0.25m, 2°	0.5m, 5°	5m, 10°	0.25m, 2°	0.5m, 5°	5m, 10°
S2DNet	MAGSAC++ [5]	0.46 (+ 55%)	0.28 (+ 80%)	0.10 (+229%)	0.62 (+ 3%)	0.41 (+ 2%)	0.31 (+ 11%)	0.70 (+ 11%)	0.44 (+ 5%)	0.30 (+ 2%)
S2DNet	RE *Min. Eq. 10*	0.32 (+ 7%)	0.20 (+ 27%)	0.08 (+165%)	**0.58** (- 4%)	0.40 (+ 1%)	0.31 (+ 13%)	0.66 (+ 6%)	0.47 (+ 13%)	0.39 (+ 31%)
S2DNet	FPR *Min. Eq. 11*	0.32 (+ 7%)	0.20 (+ 27%)	0.06 (+ 97%)	0.61 (+ 1%)	0.41 (+ 4%)	0.29 (+ 4%)	0.63 (+ 1%)	**0.41** (- 4%)	0.31 (+ 5%)
S2DNet	NRE	**0.30** (+ 0%)	**0.15** (+ 0%)	**0.03** (+ 0%)	0.60 (+ 0%)	**0.39** (+ 0%)	**0.28** (+ 0%)	**0.62** (+ 0%)	0.42 (+ 0%)	**0.29** (+ 0%)

Table 5. **NRE-based *vs*. RE-based *vs*. FPR-based pose estimators on Aachen Night [35] and InLoc [46]:** We evaluate the gain in performance of our novel NRE-based pose estimator against state-of-the-art RE-based and FPR-based pose estimators. For a fair comparison, *each method uses the same oracle nearest-neighbor database image for each query image*. Moreover, each method employs S2DNet [19] features, even our NRE-based pose estimator. For the methods that have an hyperparameter, we optimized it and report the best results. We report the error at several thresholds for translation and rotation (lower is better). The scores between brackets show the relative deterioration w.r.t. to NRE. On Aachen, there is no strong domain shift w.r.t. MegaDepth images that are used to train S2DNet, as a result the dense loss maps are accurate and our NRE-based pose estimator significantly outperforms its competitors. On InLoc, there is a strong domain shift (InLoc is an indoor dataset), as a result the dense loss maps are not very informative and our NRE-based pose estimator does not significantly outperform its competitors.

Features	Pose Estim.	Aachen Night			InLoc-DUC1			InLoc-DUC2		
		0.25m, 2°	0.5m, 5°	5m, 10°	0.25m, 2°	0.5m, 5°	5m, 10°	0.25m, 2°	0.5m, 5°	5m, 10°
S2DNet	NRE	0.30 (+ 12%)	0.15 (+ 37%)	0.03 (+ 55%)	0.60 (+ 1%)	0.40 (+ 3%)	0.28 (+ 10%)	0.63 (+ 1%)	0.42 (+ 10%)	0.30 (+ 3%)
NRE Features	NRE	**0.26** (+ 0%)	**0.11** (+ 0%)	**0.02** (+ 0%)	**0.59** (+ 0%)	**0.39** (+ 0%)	**0.25** (+ 0%)	**0.62** (+ 0%)	**0.38** (+ 0%)	**0.29** (+ 0%)

Table 6. **NRE features *vs*. S2DNet features for NRE-based pose estimators on Aachen Night [35] and InLoc [46]:** We evaluate the gain in performance of our NRE features against S2DNet [19] features using the same NRE-based pose estimator. We compare pose estimation on Aachen Night [35] and InLoc [46] images. For a fair comparison, *each method uses the same oracle nearest-neighbor database image for each query image*. We report the error at several precision thresholds for translation and rotation (lower is better). The scores between brackets show the relative deterioration w.r.t. to NRE features. On Aachen, there is no strong domain shift w.r.t. MegaDepth images that are used to train both S2DNet and our NRE feature, as a result the dense loss maps are accurate and we obtain improvements similar to the ones we obtained in our MegaDepth experiment. On InLoc, there is a strong domain shift (InLoc is an indoor dataset), as a result neither S2DNet dense loss maps nor the dense loss maps obtained using our NRE features are very informative. As a result, the pose estimated ugin NRE features is not markedly more accurate than the pose obtained using S2DNet features.

(a) Extract a 64×64 region in the dense fine descriptors around $\mathbf{q}$.

(b) Compute the dot product with the fine descriptor of the 3D point and apply a softmax to obtain $\mathtt{C}_{\text{fine}}$.

Thus by definition $\sum_{\mathbf{p} \in \mathcal{N}_{64\times 64}(\mathbf{q})} \mathtt{C}_{\text{fine}}(\mathbf{p}) = 1$.

3. $\mathtt{C}_{\text{fine}}$ corresponds to a region of size 8x8 in $\mathtt{C}_{\text{coarse}}$. Compute the sum of these 64 pixels in $\mathtt{C}_{\text{coarse}}$. We call this scalar $norm_{coarse}$.

4. Multiply $\mathtt{C}_{\text{fine}}$ by $\frac{norm_{coarse}}{64}$ to obtain $\mathtt{C}_{\text{fine norm}}$. $\mathtt{C}_{\text{fine norm}}$ is a local high-resolution version of $\mathtt{C}_{\text{coarse}}$.

5. The final local high resolution loss map is obtained classically:
$\mathtt{L}_{\text{fine}} = \min\left(\ln|\mathring{\Omega}_{\text{fine}}|, -\ln(\mathtt{C}_{\text{fine norm}})\right)$. By definition, outside of the 64×64 region, the value of the loss is $\ln|\mathring{\Omega}_{\text{fine}}|$.

D.2. Network Architectures (Sec. 6)

Coarse network architecture. The purpose of the coarse network $\mathcal{F}_{\text{coarse}}$ is to provide robust descriptors that are used to obtain a coarse pose estimate. To deal with ambiguous cases, it should leverage image context. This motivates a deep architecture with a wide receptive field and a large descriptor size. On the other hand, the network should output dense descriptors of sufficient resolution to reliably estimate a coarse camera pose. We experimentally found that an effective stride of 16 is sufficient. To satisfy these specifications, we opted for an Inception-v3 [45] backbone and modified it accordingly. We changed some kernel sizes and truncated the network at the layer Mixed-6e. In the end our final architecture has a receptive field of 927 pixels and produces dense descriptors of size $H/16 \times W/16 \times 1280$.

Fine network architecture. The purpose of the fine network $\mathcal{F}_{\text{fine}}$ is to provide discriminative high-resolution descriptors that are used to refine the coarse pose estimate. However, producing high-resolution descriptors takes a lot of memory. This motivates a deep architecture with a small receptive field and a small descriptor size. We

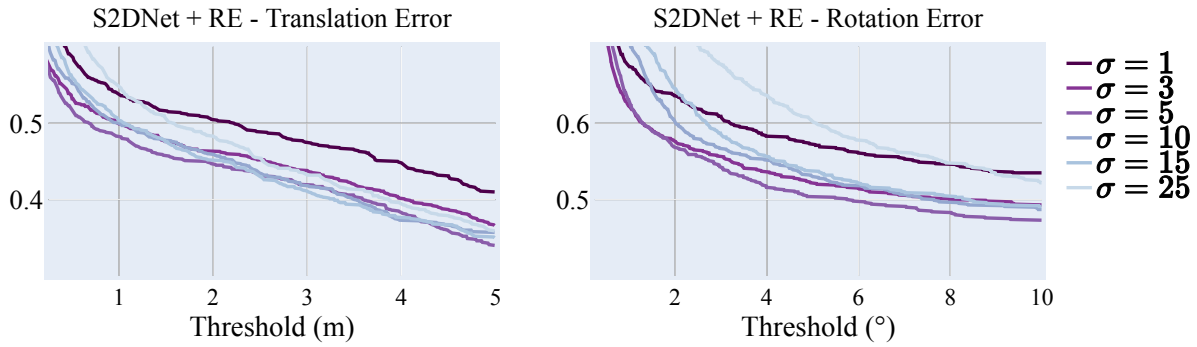

Figure 7. **Tuning the hyperparameter of an RE-based pose estimator:** We report the cumulative error curves in pose estimation (lower is better), on the hardest category of our Megadepth study, for the RE-based pose estimator that consists in minimize Eq. 10. We find that a careful hyperparameter tuning is very important. On the contrary, our novel formalism leads to a loss that does not possess any hyperparameter.

experimentally found that an effective stride of 2 is a good balance between accuracy and memory consumption. To satisfy these specifications, we opted again for a modified Inception-v3 [45] backbone. We only keep the stride of 2 at the first layer and remove any Max-Pooling layer, and we truncate the model at the Mixed-5d layer. Our final architecture has a receptive field of 43 pixels and produces dense descriptors of size $H/2 \times W/2 \times 288$.

Implementation details. The coarse network $\mathcal{F}_{\text{coarse}}$ and the fine network $\mathcal{F}_{\text{fine}}$ are trained independently. Both networks use the same training data which comes from the MegaDepth dataset [25]. As D2-Net [17], we remove scenes which overlap with the PhotoTourism [1,48] test set. We train our networks on image pairs ($\mathtt{I_S}$ and $\mathtt{I_T}$) with an arbitrary overlap.

To train $\mathcal{F}_{\text{fine}}$, we extract random crops of size 800×800 and randomly sample a maximum of 64 3D points visible in both $\mathtt{I_S}$ and $\mathtt{I_T}$. Using such large crops may seem an overkill since $\mathcal{F}_{\text{fine}}$ has a small receptive field. Let us highlight that using $C \times C$ crops allows to produce correspondence maps of size $C/2 \times C/2$ which essentially consists in comparing each source patch against C^2 target patches. Thus, even if $\mathcal{F}_{\text{fine}}$ has a small receptive field, the larger the crops during training the better the descriptors, and 800×800 is the maximum size that could fit in memory.

To train $\mathcal{F}_{\text{coarse}}$, we use entire images as inputs since the network has a very large receptive field and randomly sample a maximum of 64 3D points visible in both $\mathtt{I_S}$ and $\mathtt{I_T}$. Each network is trained using early stopping on the MegaDepth validation set. We use Adam [24] with an initial learning rate of 10^{-3} and apply a multiplicative decaying factor of $e^{-0.1}$ at every epoch.

D.3. Timing

We run all our training and experiments on a machine equipped with an Intel(R) Xeon(R) E5-2630 CPU at 2.20GHz, and an NVIDIA GeForce GTX 1080Ti GPU. The timing results reported in Tab. table:timings where obtained using a Python implementation of the previously described algorithms. Source code will be made available.

D.4. Implementation details about the RE-based *vs.* NRE-based pose estimators study

- In our RE-based *vs.* NRE-based pose estimators study, we used LO-RANSAC [14], GC-RANSAC [2] and MAGSAC++ [5] implementations provided in OpenCV 4.5.0 [1].
- We show in Fig. 7 the cumulative errors curves for several σ values when minimizing Eq. 1 on the hardest category of our Megadepth [25] study. These results stress how important a careful hyperparameter tuning is in standard RE pose estimators.
- Throughout our paper we run the coarse GNC with decreasing σ values ranging from 2.0 to 0.6. For the fine GNC, we use values between 8.0 and 0.6.

Acknowledgement

This project has received funding from the Bosch Research Foundation (*Bosch Forschungsstiftung*).

[1] https://docs.opencv.org/master/d9/d0c/group__calib3d.html

References

[1] Phototourism Challenge, CVPR 2019 Image Matching Workshop. 2019.

[2] D. Barath and J. Matas. Graph-Cut RANSAC. In *CVPR*, pages 6733–6741, 2018.

[3] D. Barath, J. Matas, and J. Noskova. MAGSAC: Marginalizing Sample Consensus. In *CVPR*, 2019.

[4] J. T. Barron. A General and Adaptive Robust Loss Function. In *CVPR*, pages 4331–4339, 2019.

[5] D. Baráth, J. Noskova, M. Ivashechkin, and J. Matas. MAGSAC++, A Fast, Reliable and Accurate Robust Estimator. In *CVPR*, pages 1301–1309, 2020.

[6] A. Benbihi, M. Geist, and C. Pradalier. ELF: EMbedded Localisation of Features in Pre-Trained CNN. In *ICCV*, pages 7940–7949, 2019.

[7] A. Bhowmik, S. Gumhold, C. Rother, and E. Brachmann. Reinforced Feature Points: Optimizing Feature Detection and Description for a High-Level Task. In *CVPR*, pages 4948–4957, 2020.

[8] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT press, 1987.

[9] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC – Differentiable RANSAC for Camera Localization. In *CVPR*, 2017.

[10] E. Brachmann and C. Rother. Learning Less Is More - 6D Camera Localization via 3D Surface Regression. *CoRR*, abs/1711.10228, 2017.

[11] E. Brachmann and C. Rother. Neural- Guided RANSAC: Learning Where to Sample Model Hypotheses. In *ICCV*, 2019.

[12] M. Bui, T. Birdal, H. Deng, S. Albarqouni, L. Guibas, S. Ilic, and N. Navab. 6D Camera Relocalization in Ambiguous Scenes via Continuous Multimodal Inference. In *ECCV*, 2020.

[13] C. Choy, J. Lee, R. Ranftl, J. Park, and V. Koltun. High-Dimensional Convolutional Networks for Geometric Pattern Recognition. In *CVPR*, pages 11227–11236, 2020.

[14] O. Chum, J. Matas, and J. Kittler. Locally Optimized RANSAC. In *DAGM-Symposium*, 2003.

[15] O. Chum, T. Werner, and J. Matas. Two-View Geometry Estimation Unaffected by a Dominant Plane. In *CVPR*, pages 772–779, 2005.

[16] D. Detone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-Supervised Interest Point Detection and Description. In *CVPR*, 2018.

[17] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A Trainable CNN for Joint Description and Detection of Local Features. In *CVPR*, 2019.

[18] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24, 1981.

[19] H. Germain, G. Bourmaud, and V. Lepetit. S2DNet: Learning Image Features for Accurate Sparse-to-Dense Matching. In *ECCV*, 2020.

[20] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm. Reconstructing the World* in Six Days *(as Captured by the Yahoo 100 Million Image Dataset). In *CVPR*, 2015.

[21] P. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:492–518, 1964.

[22] A. Kendall and R. Cipolla. Geometric Loss Functions for Camera Pose Regression with Deep Learning. In *CVPR*, pages 5974–5983, 2017.

[23] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *ICCV*, pages 2938–2946, 2015.

[24] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014.

[25] Z. Li and N. Snavely. Megadepth: Learning Single-View Depth Prediction from Internet Photos. In *CVPR*, 2018.

[26] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004.

[27] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan. Aslfeat: Learning Local Features of Accurate Shape and Localization. In *CVPR*, pages 6589–6598, 2020.

[28] Z. Lv, F. Dellaert, J. M. Rehg, and A. Geiger. Taking a Deeper Look at the Inverse Compositional Algorithm. In *CVPR*, pages 4581–4590, 2019.

[29] A. Mishchuk, D. Mishkin, F. Radenović, and J. Matas. Working Hard to Know Your Neighbor's Margins: Local Descriptor Learning Loss. In *NeurIPS*, 2017.

[30] H. Mobahi and J. W. Fisher. On the Link Between Gaussian Homotopy Continuation and Convex Envelopes. In *CVPR*, pages 43–56, 2015.

[31] K. Moo yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua. Learning to Find Good Correspondences. In *CVPR*, pages 2666–2674, 2018.

[32] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*, volume 32, pages 12405–12415. Curran Associates, Inc., 2019.

[33] I. Rocco, R. Arandjelović, and J. Sivic. Efficient Neighbourhood Consensus Networks via Submanifold Sparse Convolutions. *IEEE TPAMI*, 2020.

[34] P.-E. Sarlin, D. Detone, T. Malisiewicz, and A. Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*, 2020.

[35] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *CVPR*, 2018.

[36] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *CVPR*, 2017.

[37] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *BMVC*, 2012.

[38] J. L. Schönberger and J.-M. Frahm. Structure-From-Motion Revisited. In *CVPR*, 2016.

[39] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *ECCV*, 2016.

[40] X. Shen, C. Wang, X. Li, Z. Yu, J. Li, C. Wen, M. Cheng, and Z. He. RF-Net: An End-To-End Image Matching Network Based on Receptive Field. In *CVPR*, pages 8132–8140, 2019.

[41] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, and K. M. Yi. ACNe: Attentive Context Normalization for Robust Permutation-Equivariant Learning. In *CVPR*, pages 11286–11295, 2020.

[42] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *IEEE TPAMI*, 39(7), 2017.

[43] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate Localization and Pose Estimation for Large 3D Models. In *CVPR*, 2014.

[44] C. Sweeney, V. Fragoso, T. Höllerer, and M. Turk. Large Scale SfM with the Distributed Camera Model. In *International Conference on 3D Vision*, 2016.

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, pages 2818–2826, 2016.

[46] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. Inloc: Indoor Visual Localization with Dense Matching and View Synthesis. *CoRR*, abs/1803.10368, 2018.

[47] C. Tang and P. Tan. BA-Net: Dense Bundle Adjustment Network. In *ICLR*, 2019.

[48] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li. YFCC100M: The New Data in Multimedia Research. *Commun. ACM*, 59, 2016.

[49] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas. SOSNet: Second Order Similarity Regularization for Local Descriptor Learning. In *CVPR*, 2019.

[50] P. H. Torr and A. Zisserman. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[51] M. Tyszkiewicz, P. Fua, and E. Trulls. DISK: Learning Local Features with Policy Gradient. In *NeurIPS*, 2020.

[52] L. Von Stumberg, P. Wenzel, Q. Khan, and D. Cremers. GN-Net: The Gauss-Newton Loss for Multi-Weather Relocalization. *IEEE Robotics and Automation Letters*, 5(2):890–897, 2020.

[53] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely. Learning Feature Descriptors Using Camera Pose Supervision. In *ECCV*, 2020.

[54] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In *ECCV*, 2016.

[55] C. Zach and G. Bourmaud. Iterated Lifting for Robust Cost Optimization. In *BMVC*, 2017.

[56] C. Zach and G. Bourmaud. Descending, Lifting or Smoothing: Secrets of Robust Cost Optimization. In *ECCV*, pages 547–562, 2018.

[57] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao. Learning Two-View Correspondences and Geometry Using Order-Aware Network. In *ICCV*, 2019.