Supplementary Material: Two-view Geometry Scoring Without Correspondences

Feature Extractor		
Layer	Description	Output Shape
	Input Image	[b, 3, 256, 256]
0	Conv-BN-ReLU	[b, 128, 256, 256]
1	ResNet block 1	[b, 128, 128, 128]
2	ResNet block 2	[b, 196, 64, 64]
3	ResNet block 3	[b, 256, 32, 32]
4	ResNet block 4	[b, 256, 16, 16]
5	Up and Skip conn. w/ layer 3	[b, 256, 32, 32]
6	Conv-BN-LeakyReLU	[b, 196, 32, 32]
7	Up and Skip conn. w/ layer 2	[b, 196, 64, 64]
8	Conv-BN-LeakyReLU	[b, 128, 64, 64]

Table 1. Feature extractor architecture details. The feature extractor computes features from input images A and B at 1/4 of the input resolution. A ResNet block refers to a ResNet-18 [5] block, which is composed of 3×3 convolutions, batch normalization layers [6], ReLU activations [1], and a residual connection. The residual connection is done between the input to the block and the output. The Up and Skip conn. refers to an upsampling layer with bilinear interpolation and a skip connection between the input to the layer and the previous layer i.

Pose Error Regressor		
Layer	Description	Output Shape
	Input feature maps ($\mathbf{f}_i^A$ and $\mathbf{f}_i^B$)	[b, 128, 32, 32]
1	ResNet block 1	[b, 128, 16, 16]
2	ResNet block 2	[b, 128, 8, 8]
3	ResNet block 3	[b, 256, 4, 4]
4	ResNet block 4	[b, 512, 2, 2]
5	2D Avg. Pooling ($\mathbf{v}_i^{A\to B}$ and $\mathbf{v}_i^{B\to A}$)	[b, 512, 1, 1]
6	Max Pooling ($\mathbf{v}_i$)	[b, 512, 1, 1]
7	Conv1x1-BN-ReLU (MLP layer 1)	[b, 512, 1, 1]
8	Conv1x1-BN-ReLU (MLP layer 2)	[b, 256, 1, 1]
9	Conv1x1-BN-ReLU (MLP layer 3)	[b, 2]

Table 2. Pose error regressor architecture details. The pose error regressor block estimates the rotation (e_i^R) and the translation (e_i^t) errors for images A and B and fundamental matrix F_i. The input to the pose error regressor block is the epipolar transformed features $\mathbf{f}_i^A$ and $\mathbf{f}_i^B$. As in the feature extractor, the ResNet block refers to a ResNet-18 [5] block.

1. FSNet Architecture

Complementary to the description of FSNet from the main paper, we also include the implementation of its different blocks. Table 1 details the layers within the feature extractor block that we use for computing the features $\mathbf{f}^A$ and $\mathbf{f}^B$ from images A and B. Input images have a resolution of 256×256, and the feature extractor block outputs feature maps of size $128 \times 64 \times 64$. As seen in Table 1, the feature extractor is composed of ResNet-18 [5] blocks, where every block is based on 3×3 convolutions, batch normalization layers [6], ReLU activations [1], and a residual connection. After the ResNet blocks, we upsample the feature maps twice and create skip connections with previous layers following a UNet [10] architecture design. Please refer to Table 1 to see which layers are combined by the skip connections. A final convolution layer with a batch normalization layer and a Leaky-ReLU activation [13] generates the feature maps $\mathbf{f}^A$ and $\mathbf{f}^B$.

Once feature maps are extracted, we feed them to our transformer architecture (see Section 4.2 from the main paper). The transformer computes the transformed features ${}^{\dagger}\mathbf{f}^A$ and ${}^{\dagger}\mathbf{f}^B$, which exploit the self and cross-similarities across the feature maps. For our transformer architecture, we follow the design of the Linear Transformer [7, 12]. We use three attention layers ($N_t = 3$), where every self and cross-attention layer has eight attention heads. The transformer outputs ${}^{\dagger}\mathbf{f}^A$ and ${}^{\dagger}\mathbf{f}^B$, which are stored and reused for every F_i hypothesis.

We embed the two-view geometry into the features through an epipolar cross-attention block. The epipolar cross-attention takes ${}^{\dagger}\mathbf{f}^A$, ${}^{\dagger}\mathbf{f}^B$, and F_i to guide the attention between the two feature maps. The epipolar cross-attention layer applies cross-attention along the epipolar line. For every query point, we sample $D = 45$ positions along its corresponding epipolar line , and hence, attention is done only to the D sampled positions. Some sampling positions might be outside of the feature plane, *e.g.*, epipolar line never crosses the feature map. Thus, in those cases, we pad the positions with zeros, such that they do not contribute when computing the attended features. In the transformer Softmax, those positions will not matter as their contribution to the soft-aggregation is zero. To reduce the feature

	MAA at 10° ↑	Median (°) ↓
	R / t / max(R,t)	e_R / e_t
Fundamental		
LoFTR [12]		
MAGSAC++ [3]	0.11 / 0.03 / 0.02	27.21 / 46.53
FSNet	**0.13** / **0.05** / **0.03**	**20.98** / **38.42**
w/ Corresp. filter	0.11 / 0.03 / 0.02	26.48 / 46.28
w/ Candidate filter	**0.13** / 0.04 / **0.03**	21.98 / 40.34
SIFT [8]		
MAGSAC++ [3]	0.08 / 0.03 / 0.02	87.26 / 50.02
FSNet	0.08 / 0.03 / 0.02	**36.78** / **43.96**
w/ Corresp. filter	0.08 / 0.03 / 0.02	47.21 / 47.94
w/ Candidate filter	**0.09** / 0.03 / 0.02	45.35 / 46.47
Essential		
LoFTR [12]		
MAGSAC++ [3]	0.17 / 0.09 / **0.07**	22.50 / 37.79
FSNet	**0.20** / 0.10 / 0.07	**17.56** / **31.79**
w/ Corresp. filter	0.17 / 0.09 / 0.07	21.98 / 37.64
w/ Candidate filter	**0.20** / **0.11** / **0.08**	18.88 / 34.02
SIFT [8]		
MAGSAC++ [3]	0.14 / 0.06 / 0.05	73.28 / 46.06
FSNet	0.15 / 0.07 / **0.06**	**27.73** / **38.74**
w/ Corresp. filter	0.15 / 0.06 / 0.05	42.03 / 43.11
w/ Candidate filter	**0.16** / **0.08** / **0.06**	38.05 / 41.38

Table 3. Integrating FSNet with LoFTR [12] and SIFT [8]. MAA at 10° and Median error (°) results for FSNet and MAGSAC++ on the fundamental and essential matrix estimation task on the ScanNet indoor dataset. As a reference, LoFTR detects fewer than 100 correspondences on 6.5% of the image pairs. Meanwhile, the test set based on SIFT correspondences results in 3,319 (0-100) and 1,681 (100-Inf) image pairs. We observe that when there are lower quality of correspondences, FSNet comes out ahead, *i.e.*, FSNet (alone) returns the lowest pose errors.

map size, which contributes towards faster processing time, we query points every two positions. The epipolar cross-attention layer outputs $\mathbf{f}_i^A$ and $\mathbf{f}_i^B$ at 1/8 of the input image resolution ($128 \times 32 \times 32$).

Table 2 shows the details of the pose error regressor block. The pose error regressor takes the epipolar attended features ($^\dagger\mathbf{f}^A$, $^\dagger\mathbf{f}^B$) and predicts the translation and rotation errors (e_i^t and e_i^R) associated with F_i. Similar to the feature extractor, the pose error regressor uses ResNet-18 blocks to process the features. After processing the features, a 2D average pooling is applied to create $\mathbf{v}_i^{A\to B}$ and $\mathbf{v}_i^{B\to A}$. To enforce image order-invariance, we merge the two vectors with a Max-Pooling operator. A final MLP uses the output of the Max-Pool v_i to predict the pose errors.

2. FSNet with LoFTR and SIFT correspondences

FSNet is trained using hypotheses generated with SP-SG [4,11] correspondences and MAGSAC++ [3]. While FSNet does not rely on correspondences to do scoring, the hypothesis pool is generated from correspondences. In the main paper (Table 4), we show experiments where FSNet, which is trained with hypotheses generated by SP-SG, is used to score hypotheses generated by SIFT features, outperforming the MAGSAC++ scoring function, and showing the generalization capability of FSNet.

We then extend the previous experiment and show results of FSNet generalizing to hypotheses generated by a different correspondence estimation method. We choose LoFTR [12], another state-of-the-art matching network. We use Kornia [9] library to compute LoFTR correspondences. Table 3 shows the results of MAGSAC++ and FSNet combined with LoFTR. As a reference, we also include SP-SG results.

We observe that (i) LoFTR performance is lower than SP-SG, and (ii) MAGSAC++ alone or in combination is struggling, leaving FSNet (alone) as the winner. We believe that one possible cause for LoFTR's lower performance is the distribution of our test set, which uses image pairs with very low image overlap (10%-40%), and hence, it is different from the image pairs used for LoFTR training. Besides LoFTR results, we also show in Table 3 that FSNet can be paired with *e.g.* SIFT. SIFT matches are filtered with the mutual nearest neighbor check and Lowe's ratio test [8]. Although the distribution of hypotheses generated by SIFT is potentially different, FSNet ranks them successfully achieving similar mAA scores as MAGSAC++, while reducing the median pose error.

3. More Correspondences for Difficult Image Pairs

In the main paper's Section 3, we mention that loosening the filtering criteria of SuperGlue, and thus increasing the number of correspondences provided to MAGSAC++, does not lead to improvements in scores.

Indeed, SuperGlue filters correspondences by considering the matching confidence of the correspondences, where the threshold is 0.2. However, given that the number of correspondences impacts the performance of correspondence-based scoring methods (Figure 3 of the main paper), would more correspondences improve the estimation of the fundamental or essential matrices? To investigate this, we varied the filtering threshold of SuperGlue to increase the number of correspondences that go into the RANSAC loop when needed. So, we compile a list of image pairs that initially have ≤ 100 correspondences. These are image pairs in our ScanNet test set that are used to report scores of 0-100 splits in tables 2, 3, and 4 of the main paper. Then, we lower the threshold progressively by steps of 0.04 until either; we obtain more than 100 correspondences, or we reduce the threshold to 0.0 (hence, we will use all possible matches). We refer to this SuperGlue as SP-SG*, and show in Table 4

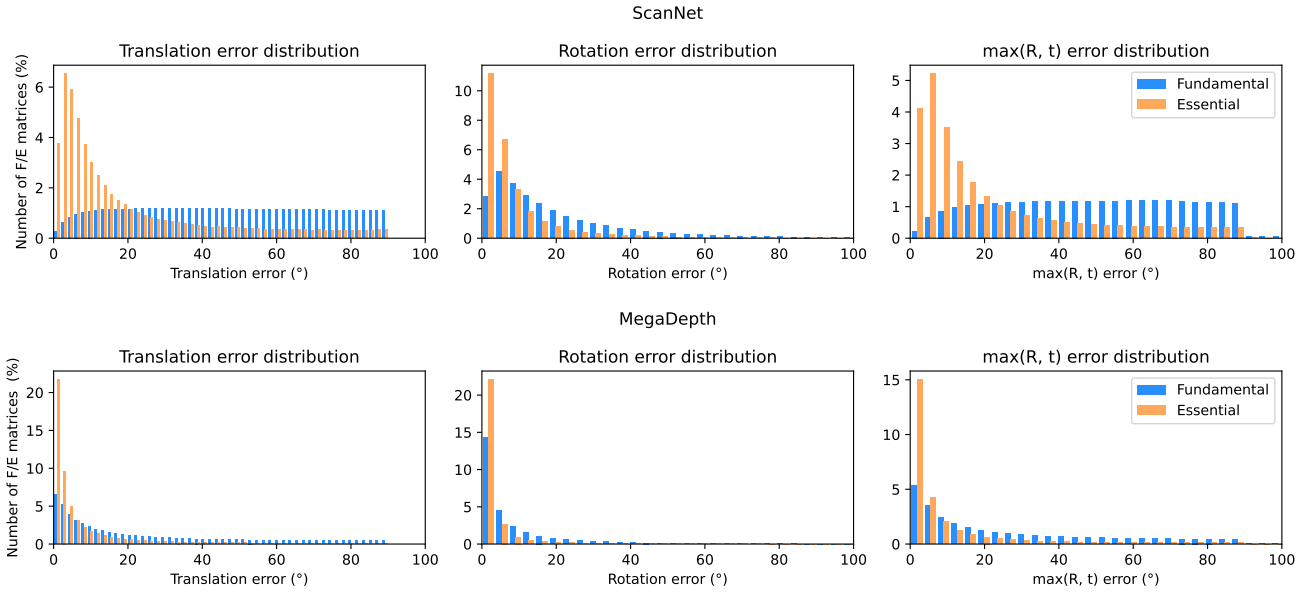


Figure 1. Fundamental vs essential matrix error distributions. The figure shows the error distributions (°), translation, rotation, and the maximum of both, for the generated fundamental and essential matrices with SP-SG correspondences. We observe that essential matrices have a higher population on the low error regime, *i.e.*, matrices with pose error below 20°. Meanwhile, fundamental matrices show a wider range of errors, especially in the indoor scenario, where correspondences do not provide enough reliability for accurate two-view geometry estimation. This observation leads us to train FSNet for either the task of fundamental or essential matrix estimation.

	MAA at 10° ↑	**Median (°) ↓**
	R / t / max(R,t)	e_R / e_t
Fundamental		
SP-SG* + MAGSAC++	0.38 / 0.18 / 0.14	6.22 / 16.50
SP-SG + MAGSAC++	0.38 / 0.17 / 0.14	6.35 / 17.89
FSNet	0.36 / 0.21 / 0.15	6.52 / 12.05
w/ Corresp. filter	0.39 / 0.22 / 0.16	6.09 / 11.59
w/ Candidate filter	**0.43 / 0.23 / 0.19**	**5.38 / 11.39**
Essential		
SP-SG* + MAGSAC++	0.38 / 0.24 / 0.19	6.38 / 11.30
SP-SG + MAGSAC++	0.40 / 0.26 / 0.21	5.95 / 10.96
FSNet	**0.44** / 0.28 / 0.22	5.13 / 8.95
w/ Corresp. filter	**0.44 / 0.29 / 0.23**	**5.07 / 8.75**
w/ Candidate filter	**0.44** / 0.28 / **0.23**	5.14 / 9.48

Table 4. SuperGlue with more correspondences in the indoor ScanNet dataset. SP-SG* refers to SP-SG with dynamic matching confidence threshold, which is reduced in order to either obtain at least 100 correspondences or the maximum correspondences that SP-SG provides. It can be seen that naively increasing the number of correspondences does not lead to improved results.

its evaluation with MAGSAC++. As mentioned previously, an increased number of correspondences does not consistently improve scores for inlier counting baselines. Indeed, MAA scores are almost the same for the fundamental matrix estimation task and slightly worse for essential matrix estimation task. The median errors are lower for the fundamental matrix estimation when more correspondences are used, but the errors increase for essential matrix estimation. Note that the numerical results have elements of randomness to them, as a different number of correspondences produces different pools of 500 random hypotheses.

4. Distribution of F and E Hypotheses

When evaluating FSNet in the fundamental or essential matrix estimation task, we observe that specializing FSNet for a specific task was more effective than training the architecture to solve both tasks at the same time (Table 2 from main paper). This behaviour is explained by looking into Figure 1. The figure shows the error distributions of the generated fundamental and essential matrices of our validation set. For completeness, we report the translation and rotation error separately, as well as the distributions for ScanNet and MegaDepth datasets.

We observe that the distributions of the errors are different when estimating fundamental or essential matrices, where fundamental matrices tend to have a wider range of pose errors, thus, making it more difficult to select accurate estimates. This observation is also in line with the results of Table 2 from the main paper, where we show that FSNet was more effective when trained for a specific task, instead of using matrices from both distributions (F + E).

	0-100		100-Inf		All	
	MAA at 10° ↑	**Median (°) ↓**	**MAA at 10° ↑**	**Median (°) ↓**	**MAA at 10° ↑**	**Median (°) ↓**
	R / t / max(R, t)	e_R / e_t	R / t / max(R, t)	e_R / e_t	R / t / max(R, t)	e_R / e_t
Fundamental						
MAGSAC++ [3]	0.41 / 0.17 / 0.14	5.55 / 16.44	0.76 / 0.33 / 0.32	1.77 / 7.74	0.61 / 0.26 / 0.24	2.71 / 10.49
FSNet (F)	0.41 / 0.22 / 0.17	5.62 / 10.84	0.62 / 0.24 / 2.22	3.02 / 11.07	0.53 / 0.23 / 0.19	3.87 / 10.91
w/ Corresp. filter	0.41 / 0.22 / 0.17	5.62 / 10.84	0.76 / 0.33 / 0.32	**1.77** / 7.74	0.61 / 0.28 / 0.25	2.82 / 8.91
w/ Candidate filter	**0.49 / 0.26 / 0.21**	**4.36 / 10.47**	**0.78 / 0.38 / 0.36**	1.79 / **6.06**	**0.65 / 0.32 / 0.30**	**2.46 / 7.58**
Essential						
MAGSAC++ [3]	0.42 / 0.26 / 0.21	5.39 / 10.96	0.76 / 0.41 / 0.40	1.72 / 5.58	0.61 / 0.34 / 0.31	2.63 / 7.41
FSNet (E)	0.47 / 0.28 / 0.22	4.57 / **8.85**	0.72 / 0.35 / 0.33	2.22 / 6.80	0.61 / 0.32 / 0.28	3.06 / 7.53
w/ Corresp. filter	0.47 / 0.28 / 0.22	4.57 / **8.85**	0.76 / 0.41 / 0.40	1.72 / 5.58	0.64 / 0.35 / 0.32	2.66 / 6.82
w/ Candidate filter	**0.48 / 0.29 / 0.24**	**4.47** / 9.26	**0.79 / 0.45 / 0.43**	**1.69 / 4.97**	**0.66 / 0.38 / 0.35**	**2.43 / 6.36**

Table 5. Fundamental and essential matrix estimation on SuperGlue's [11] test set of ScanNet. We compute the MAA at 10° and Median (°) metrics in the SuperGlue's test split of ScanNet, and divide the pairs of images based on the number of SP-SG correspondences. The split results in 649 (0-100) and 851 (100-Inf) image pairs. We see that when the number of correspondences is small (0-100), FSNet outperforms MAGSAC++ in both, fundamental and essential matrix estimation. In the overall split (All), the best results are obtained when combining FSNet and MAGSAC++ hypothesis scores.

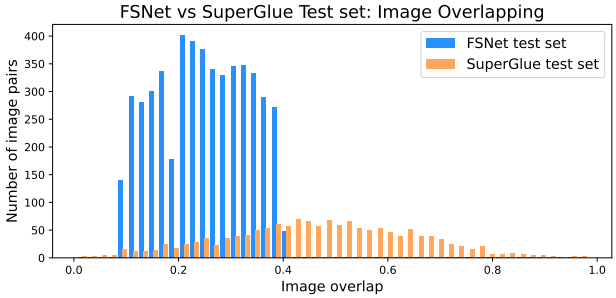


Figure 2. Overlapping distribution of the image pairs of FSNet and SuperGlue [11] test sets. FSNet focuses on image pairs where current correspondence-scoring methods struggle. Therefore, to tackle such cases, we generate FSNet training, validation, and test sets with low visual overlapping images, where correspondences are few and then, in many cases, not reliable.

5. SuperGlue Test Set

Results in the main paper are reported on our own test set for Scannet. We generate this test set split such that image pairs have a distribution of image overlaps that focuses on hard-to-handle image pairs (please see blue bars in Figure 2). However, SuperGlue also published image pairs that they used for evaluation. For completeness and easy comparison, we provide evaluations of FSNet on SuperGlue's test set.

We benchmark our performance on the standard SuperGlue test set [11], with results shown in Table 5. FSNet can be seen as helpful there, but that table's scores are dominated by "easy pairs": we see that the regular SuperGlue test set has many image pairs with high overlap scores. That SuperGlue test set has limited exposure to scoring-failure cases, as can be seen from the overlapping statistics of the SuperGlue test set in Figure 2.

We observe the same trend in Table 5 and Table 2 from the main paper. FSNet improves over MAGSAC++ in the 0-100 split, both for fundamental and essential matrix estimation tasks. Similarly, the combined approaches with FSNet provide the best scores.

6. FSNet Trained with Cross Entropy Loss

Experiments in the main paper's Section 5.3 (Table 4) show FSNet trained with the binary cross-entropy loss. We treat the hypotheses ranking problem as a classification problem. Hypotheses with pose error $< 10°$ were labeled as "correct" and others as "incorrect", so

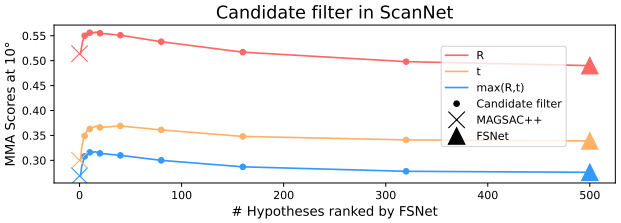
$$y_i = \begin{cases} 1, & \text{if } \max(e_i^R, e_i^t) < 10° \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

This means that out of 500 hypotheses generated for an image pair, multiple hypotheses can be labeled as "correct". As we are interested in ranking hypotheses during scoring, we can use the network's confidence in the predicted binary decision to provide ranking among "correct" hypotheses. Furthermore, we are not interested in the relative ranking of "incorrect" hypotheses. To reflect these nuances, we incorporate the network's predicted confidence in the loss function. Inspired by Barath *et al.*'s [2] loss function, we modify the cross entropy loss to be

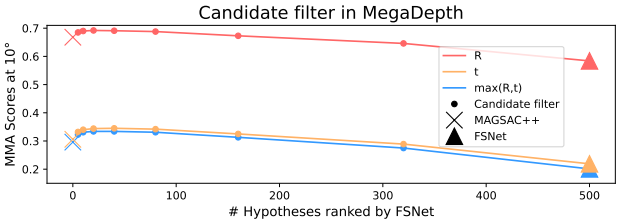
$$\mathcal{L} = -\left(1 + f(\mathbf{F}_i)\right)^w \left[y_i \log f(\mathbf{F}_i) + (1 - y_i) \log(1 - f(\mathbf{F}_i))\right], \tag{2}$$

	Unimodal (39.16%)		**Multimodal** (60.84%)			
	FSNet	MAGSAC++	FSNet ($e < 10°$)	MAGSAC++ ($e < 10°$)	Both with $e < 10°$	Both with $e \geq 10°$
%	69.40	30.60	14.34	3.55	16.67	65.43

Table 6. Distribution of top ranking hypotheses by MAGSAC++ scores in ScanNet validation set. We report the percentage of times we find an unimodal or multimodal distribution among the top 5 hypotheses returned by MAGSAC++. The criteria to determine if a pool of hypotheses is unimodal or multimodal is based on their pairwise distances. If the difference between the minimum and maximum distance is above $10°$, then we indicate it as a multimodal, otherwise, we mark the pool as unimodal. Besides the distinction between unimodal or multimodal, we also indicate which method was able to select the best hypothesis (unimodal), or which method was able to select a valid hypothesis (a hypothesis with a pose error below $10°$ *w.r.t.* the ground-truth pose).



(a) FSNet with candidate filter in ScanNet validation set.



(b) FSNet with candidate filter in MegaDepth validation set.

Figure 3. Combination of FSNet and MAGSAC++ based on the proposed candidate filter in the fundamental matrix estimation task. When using the candidate filter, FSNet only scores the top-k MAGSAC++ hypotheses. If the number of hypotheses to score is 1, it refers to original MAGSAC++, while 500 hypotheses corresponds to FSNet alone. As a reference, we also indicate MAGSAC++ ($\times$) and FSNet ($\blacktriangle$) scores.

where $f(\mathbf{F}_i)$ is FSNet network's confidence of $\mathbf{F}_i$ being a correct hypothesis, and the $w = 2$ is the weight of the networks' confidence in the loss function. Low values of $f(\mathbf{F}_i)$ means that the weighting coefficient $(1 + f(\mathbf{F}_i))^w$ is low, while high confidence values of $f(\mathbf{F}_i)$ provide higher weighting to the cross entropy loss.

7. Filtering hypotheses with MAGSAC++

As mentioned in the main paper, we can discard not promising hypotheses by looking at their MAGSAC++ scores. This filtering approach exploits the useful information that inlier counting brings to well defined set of correspondences, while also removes easy to detect outliers with such heuristics. Besides cleaning the pool of hypotheses, it also provides a speedup opportunity since FSNet only needs to be run for a subset of fundamental/essential matrices.

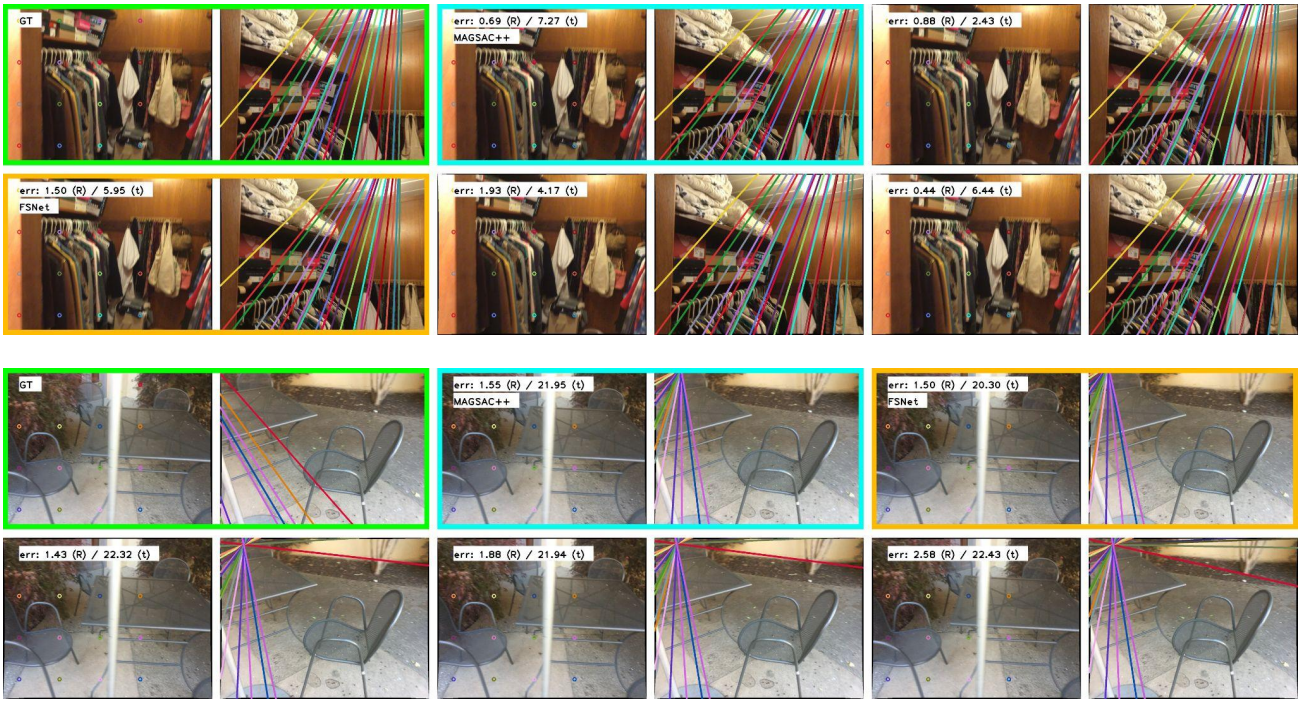
In Table 6, we analyze the top scoring hypotheses returned by MAGSAC++. Specifically, we look into the top 5 MAGSAC++ models, and analyze whether they present a multimodal or unimodal distribution. We define the distribution as unimodal if the minimum and maximum distance between the hypotheses (within the top 5) is below $10°$. In the unimodal scenario, we further look if most accurate hypothesis was returned either by MAGSAC++ or FSNet. Similarly, in the multimodal scenario (there are hypotheses with more than $10°$difference), we identify whether 1) both methods return a valid hypothesis (with a pose error below $10°$ *w.r.t.* ground-truth pose), 2) an incorrect hypothesis ($e > 10°$) was selected by both methods, 3) only FSNet or 4) MAGSAC++ selected a correct hypothesis. We observe that FSNet returns more accurate hypotheses than MAGSAC++, in both unimodal and multimodal distribution scenarios. Moreover, in Figure 3, we show the impact of varying the number of hypotheses to score by FSNet. On the left side, when only using 1 hypothesis to score, results correspond to MAGSAC++ method, meanwhile, on the right end, when using all 500 possible hypotheses, results indicate original FSNet results. We see in the ScanNet and MegaDepth datasets that even refining the score of a few hypotheses brings improvements to MAGSAC++. Moreover, in Figure 4, we show examples of different top scoring hypotheses returned by MAGSAC++, and indicate if they belong to an unimodal or multimodal distribution.

References

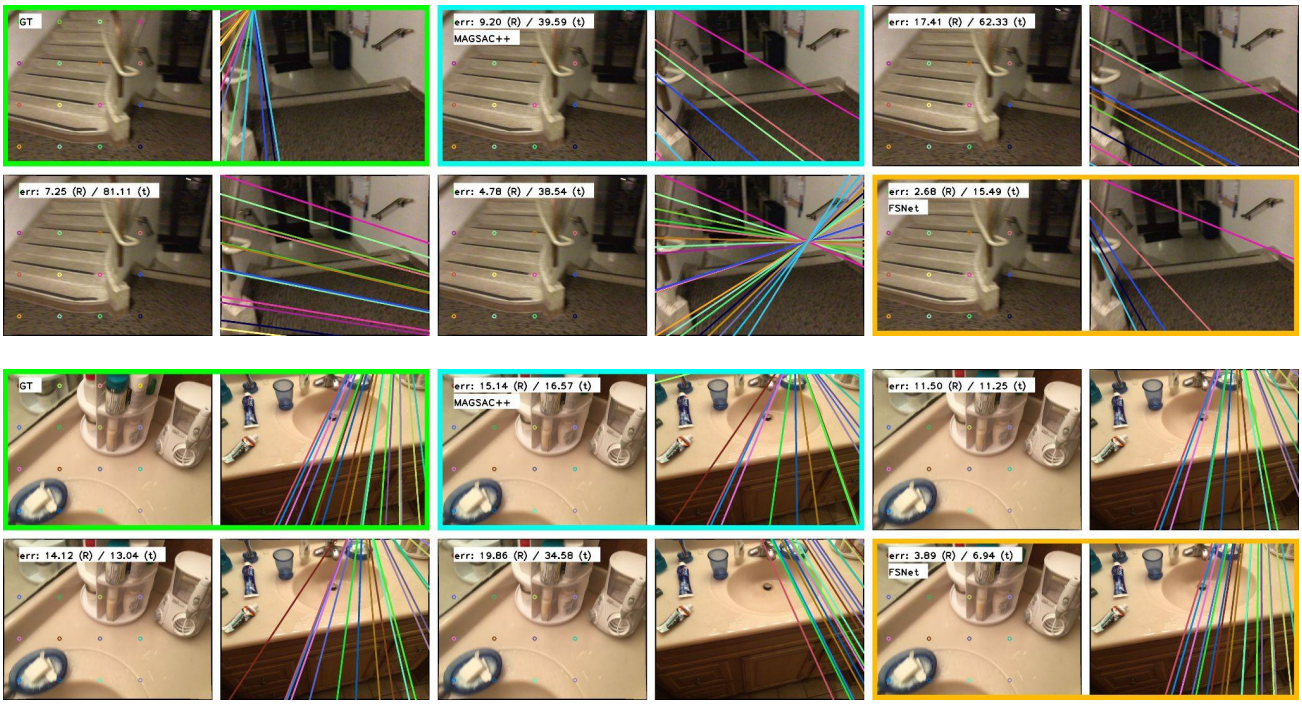
[1] Abien Fred Agarap. Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375*, 2018. 1

[2] Daniel Barath, Luca Cavalli, and Marc Pollefeys. Learning to find good models in RANSAC. In *CVPR*, pages 15744–15753, 2022. 4

[3] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1304–1312, 2020. 2, 4



(a) Unimodal distribution.



(b) Multimodal distribution.

Figure 4. Example of top scoring hypotheses returned by MAGSAC++. (a) Shows unimodal examples, where all returned hypotheses are similar, while (b) returns hypotheses with different distributions. Ground-truth is boxed in green, MAGSAC++ selection in blue, and FSNet in orange.

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 1

[7] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 1

[8] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 2

[9] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for PyTorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3674–3683, 2020. 2

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[11] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 4

[12] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 1, 2

[13] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 1