Extreme Rotation Estimation using Dense Correlation Volumes —Supplemental Material—

Ruojin Cai[1] Bharath Hariharan[1] Noah Snavely[1,2] Hadar Averbuch-Elor[1,2]
[1]Cornell University [2]Cornell Tech

Contents

1. Implementation Details

1.1. Network architecture

Our approach follows an encoder-decoder architecture as shown in Figure 2 in the main paper. We adopt a Deep-Residual-Unet(ResUNet) [11] architecture as the shared-weight Siamese **encoder**, which downsamples and upsamples the input image and outputs $32 \times H/4 \times W/4$ embedded feature maps. The image is first passed to a 7×7, stride-2 convolution layer, followed by three pre-activation residual blocks [7], each of which downsamples by 2. Reaching the lowest resolution, the feature maps are upsampled by two "up-convolution" layers. For each layer, the scale is set to 2 and a 3×3 convolution is used.

We then efficiently compute pairwise extracted feature maps using matrix multiplication and output a **4D correlation volume** as $H/4 \times W/4 \times H/4 \times W/4$. The following **decoders**, as shown in Figure 1, processes the 4D correlation volume, which is reshaped as $(H/4 \times W/4) \times H/4 \times W/4$, with two pre-activation residual blocks and uses two fully-connected layers to map to a 360-dim distribution for each angle. In total, our model contains $\sim$19M parameters, and the regression baseline contains $\sim$23M parameters.

1.2. Experimental setting

For all the experiments, we use the Adam optimizer ($\beta_1 = 0.5, \beta_2 = 0.9$). The initial learning rate for the model when using a classification loss is 5×10^{-4} (and 1×10^{-4} for the models trained with a regression loss). All models are trained over 500k iterations with a batch size of 20, using a linear decay strategy, where the learning rate drops starting from 250k iteration and ends up at 5×10^{-6} for model with classification loss (and 1×10^{-6} for models trained using a regression loss). All the models are trained and evaluated using the same training strategy. The training time is about two days with one GEFORCE RTX 2080 Ti GPU.

1.3. Baselines

In this section, we provide more details on the baselines, including ones not in the main paper due to space constraints.

SIFT-based relative rotation estimation. First, we detect local features in images of size 256×256 with SIFT [8], and then features are matched across images by fitting a model using RANSAC [6]. The minimal number of inlier pairs is set to 10. We decompose a rotation matrix with a 2-point algorithm [1] for image pairs from the same panorama and an essential matrix for image pairs with translation using RANSAC [6]. We use the publicly available OpenCV implementation for all of these steps (except for the 2-point algorithm for which we use our own implementation).

Learning-based feature matching. We use the following pretrained networks:

- D2-Net [4]. Detect-and-describe (D2) networks use an ImageNet pretrained VGG16 [10] network as the feature extraction network. The extracted feature maps are used to detect keypoint with hard feature detections and serve as local descriptors at the same time.

- SupetPoint [3]. SuperPoint uses a VGG-style [10] encoder, and passes the feature maps to two separate decoders for interest point detection and description. The head of the interest point decoder adopts sub-pixel convolution [9], and the head of the descriptor decoder uses a model similar to Universal Correspondence Networks [2]. Both decoders use non-learned upsampling modules to compute the output.

These networks detect interest points in images and generate corresponding dense feature descriptors. We then estimate the rotation matrix using a model fitting technique following the procedure described above.

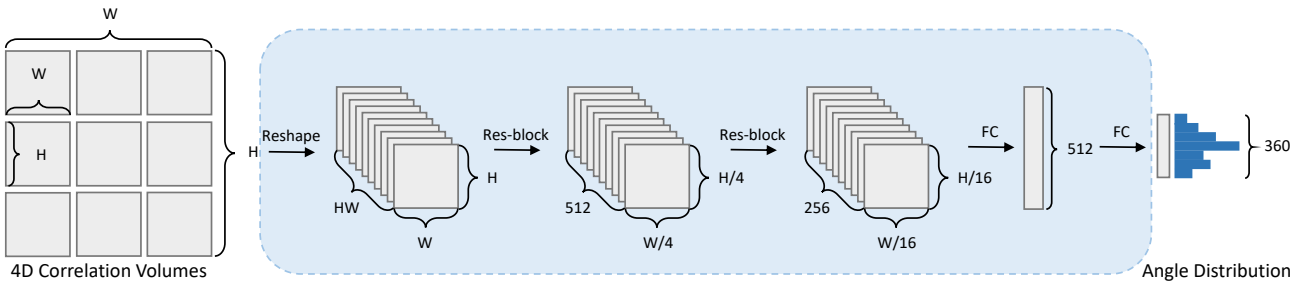


Figure 1. **Architecture of decoder.** The 4D correlation volume is reshaped and processed with two pre-activation residual blocks, and eventually mapped by two fully-connected layers to a 360-dim distribution for each angle.

End-to-end relative rotation regression. We use the following regression baselines:

- Zhou *et al*. [12], hereby denoted by *Reg6D*. Image features are concatenated and fed to a regression model predicting a continuous representation in 6D. We report two different models: (1) *Reg6D-128*, that adopts the same network architecture and pretrained weights as ours, and the same input resolution as ours, 128×128, and (2) *Reg6D*, where we replaced the ResUNet architecture of the encoder with ResNet, that outputs $1024 \times H/64 \times W/64$ embedded feature maps, as we found this modification yields improved performance. In addition, the resolution of the input images for this model is 256×256.
- En *et al*. [5], hereby denoted by *Reg4D*. The learning-based method proposed in En *et al*. concatenates the pairwise images features and then regresses them to predict a quaternion representation. We adopt the same network architecture as for the Reg6D model, and the resolution of the input images for the baselines is also set to 256×256.
- *RegEuler*. We implemented an additional regression baseline built on our network architecture, which regresses the concatenated feature to an Euler angle representation. For this additional baseline, we report results for input images of size 128×128, which is the input resolution used by our model.

2. Additional Experimental Results

2.1. Comparison to all baselines

We compare our approach to all the baselines described in the previous section and report the results in Table 1. These results further validate that our approach consistently outperforms alternative techniques. Note that the strongest baselines are reported also in the main paper.

We show the number of successful image pairs for SIFT, D2-Net and SuperPoint in Table 2. We observe a significant drop in SuperPoint's ability to find enough good matches to estimate the rotation matrix when the image pairs have larger rotations and translations.

Also, we can see that using a smaller input image (128×128 v.s. 256×256) negatively affects the performance of Reg6D in non-overlapping cases, increasing the gap with our models (which are trained using images of size 128×128) from a mean error of $29.82°$ to $48.37°$ on SUN360 and from $22.71°$ to $29.01°$ on StreetLearn.

2.2. Statistical analysis

We conduct a statistical analysis of the geodesic error for the relative rotations. As illustrated in the histograms shown in Figure 2, most errors fall in the first bin ($[0°, 10°]$), with the frequency of errors below $10°$ being significantly larger than in the baselines. As shown in the percentile of errors of our method in Figure 3, approximately $\%80$ of the errors are below $5°$. In the histogram and percentiles of the interior datasets (InteriorNet and SUN360), there's a distinct peak of error at $90°$, indicating that our method is confused by certain scenes. From Figure 4, we can see the errors at $90°$ and $180°$ mainly stem from non-overlapping image pairs (Figure 5 that shows failure cases also demonstrates this).

2.3. Roll angle experiments

We conduct a study of the influence of small roll angles by feeding our models (trained without roll) with image pairs containing small roll angles. Results are reported in Table 3. As illustrated in the table, the mean geodesic errors over pairs with up to $5°$ roll increase by up to $2°$ (across all overlap levels), demonstrating that adding roll at test time does not break the model, but rather leads to a modest increase in error.

2.4. Pitch angle prediction from a single image

We evaluate to what extent pitch angles can be estimated from a single image by inputting the same image twice into our framework. The mean error (over pitch angle) over the StreetLearn dataset is $0.75°$, with almost all images having

Overlap	Method	InteriorNet			InteriorNet-T			SUN360			StreetLearn			StreetLearn-T		
		Avg(°↓)	Med(°↓)	10°(%↑)	Avg(°↓)	Med(°↓)	10°(%↑)	Avg(°↓)	Med(°↓)	10°(%↑)	Avg(°↓)	Med(°↓)	10°(%↑)	Avg(°↓)	Med(°↓)	10°(%↑)
Large	SIFT* [8]	6.09	4.00	84.86	7.78	2.95	55.52	5.46	3.88	93.10	5.84	3.16	91.18	18.86	3.13	22.37
	D2-Net* [4]	8.27	3.78	69.48	14.19	9.38	15.82	10.48	4.22	69.46	12.85	4.42	54.12	8.76	6.73	1.32
	SuperPoint* [3]	5.40	3.53	87.10	5.46	2.79	65.97	4.69	3.18	92.12	6.23	3.61	91.18	6.38	1.79	16.45
	RegEuler-o	4.89	3.52	91.56	8.36	4.64	80.00	6.57	5.16	85.22	3.99	3.20	95.29	15.15	7.13	62.50
	RegEuler	11.11	7.92	60.05	20.98	14.59	33.43	14.19	10.77	46.31	27.47	17.59	23.53	46.89	32.36	11.84
	Reg4D [5]-o	4.95	3.47	91.56	9.58	7.00	69.25	7.86	5.89	76.85	3.14	2.67	98.82	11.73	6.35	74.34
	Reg4D [5]	13.83	10.26	48.88	26.05	16.84	22.99	36.02	24.23	16.26	20.57	13.05	40.00	41.44	28.27	20.39
	Reg6D [12]-o-128	7.48	5.24	75.19	14.58	10.97	46.27	11.97	8.11	57.64	6.00	5.20	85.29	13.66	8.24	58.55
	Reg6D [12]-128	14.04	9.06	53.10	31.96	21.19	22.99	40.19	33.41	8.87	14.09	8.63	56.47	31.01	19.14	26.97
	Reg6D [12]-o	5.43	3.87	87.10	10.45	6.91	67.76	7.18	5.79	81.28	3.36	2.71	97.65	12.31	6.02	69.08
	Reg6D [12]	9.05	5.90	68.49	17.00	11.95	41.79	16.51	12.43	40.39	11.70	8.87	58.24	36.71	24.79	23.03
	Ours-o	**1.53**	1.10	**99.26**	**2.89**	**1.10**	**97.61**	**1.00**	**0.94**	**100.00**	**1.19**	**1.02**	**99.41**	9.12	2.91	**87.50**
	Ours	1.82	**0.88**	98.76	8.86	1.86	93.13	1.37	1.09	99.51	1.52	1.09	**99.41**	24.98	2.48	78.95
Small	SIFT* [8]	24.18	8.57	39.73	18.16	10.01	18.52	13.71	6.33	56.77	16.22	7.35	55.81	38.78	13.81	5.68
	D2-Net* [4]	14.21	8.50	3.42	–	–	0.00	25.49	9.17	4.51	41.35	18.43	1.66	67.27	67.27	0.00
	SuperPoint* [3]	16.72	8.43	21.58	11.61	5.82	11.73	17.63	7.70	26.69	19.29	7.60	24.58	6.80	6.85	0.95
	RegEuler-o	15.37	7.88	59.59	19.27	7.41	64.51	14.88	8.92	54.14	8.24	4.29	86.71	20.41	9.96	50.47
	RegEuler	28.58	17.77	31.85	32.87	21.96	18.52	32.28	25.41	12.41	52.08	37.35	4.32	58.43	47.27	7.57
	Reg4D [5]-o	16.62	8.46	57.88	24.77	12.42	41.67	15.02	9.81	51.88	6.55	4.15	91.03	14.56	7.46	65.93
	Reg4D [5]	32.86	22.41	15.75	41.52	28.46	10.80	66.59	57.84	0.38	38.39	24.77	10.63	50.42	34.07	15.46
	Reg6D [12]-o-128	22.45	12.57	38.70	31.15	16.58	27.47	23.17	15.08	24.06	9.12	5.71	82.06	20.70	10.69	45.43
	Reg6D [12]-128	36.37	23.81	18.84	54.24	39.94	14.81	67.97	60.91	0.38	24.03	15.13	30.56	41.07	28.33	17.03
	Reg6D [12]-o	17.83	9.61	51.37	21.87	11.43	44.14	18.61	11.66	39.85	7.95	4.34	87.71	15.07	7.59	63.41
	Reg6D [12]	25.71	15.56	33.56	42.93	28.92	23.15	42.55	32.11	9.40	24.77	15.11	30.56	46.61	34.33	13.88
	Ours-o	6.45	1.61	95.89	**10.24**	**1.38**	**89.81**	**3.09**	**1.41**	**98.50**	**2.32**	**1.41**	**98.67**	**13.04**	3.49	**84.23**
	Ours	**4.31**	**1.16**	**96.58**	30.43	2.63	74.07	6.13	1.77	95.86	3.23	**1.41**	98.34	27.84	**3.19**	74.76
None	SIFT* [8]	109.30	92.86	0.00	93.79	113.86	0.00	127.61	129.07	0.00	83.49	90.00	0.38	85.90	106.84	0.38
	D2-Net* [4]	–	–	0.00	–	–	0.00	171.21	171.21	0.00	–	–	0.00	–	–	0.00
	SuperPoint* [3]	120.28	120.28	0.00	–	–	0.00	149.80	165.24	0.00	–	–	0.00	–	–	0.00
	RegEuler	52.95	36.03	7.87	55.73	42.04	9.97	70.93	59.43	5.46	55.92	41.23	7.56	61.04	48.79	9.04
	Reg4D [5]	62.04	48.92	4.59	59.85	48.81	4.99	80.08	72.78	1.32	46.19	32.74	9.26	55.70	39.70	9.79
	Reg6D [12]-128	64.59	49.80	5.90	79.86	71.60	5.87	83.29	75.08	0.56	34.78	23.16	17.77	50.96	36.50	9.23
	Reg6D [12]	48.36	32.93	10.82	60.91	51.26	11.14	64.74	56.55	3.77	28.48	18.86	24.39	49.23	35.66	11.86
	Ours	**37.69**	**3.15**	**61.97**	**49.44**	**4.17**	**58.36**	**34.92**	**4.43**	**61.39**	**5.77**	**1.53**	**96.41**	**30.98**	**3.50**	**72.69**
All	SIFT* [8]	13.68	5.04	45.80	12.24	5.69	24.60	18.12	5.02	34.00	17.29	5.53	32.50	36.00	6.03	5.40
	D2-Net* [4]	8.56	3.95	29.00	14.19	9.38	5.30	13.80	4.62	15.30	16.41	5.38	9.70	23.39	11.87	0.20
	SuperPoint* [3]	8.19	4.08	41.40	6.62	3.38	25.90	11.09	4.00	25.80	11.52	4.80	22.90	6.42	2.62	2.80
	RegEuler	28.97	15.53	35.90	36.68	22.70	20.60	49.13	31.44	15.60	49.93	35.08	9.30	58.06	45.54	9.00
	Reg4D [5]	34.09	19.59	25.70	42.59	28.35	12.90	67.55	55.40	4.10	39.48	25.76	14.90	51.86	36.89	13.20
	Reg6D [12]-128	35.98	18.65	28.70	55.51	36.90	14.50	70.46	57.87	2.20	28.03	17.11	28.20	44.79	31.00	14.40
	Reg6D [12]	25.90	13.02	40.70	40.38	23.35	25.30	49.05	34.37	12.70	24.51	15.31	32.00	46.50	33.14	29.90
	Ours	**13.49**	**1.18**	**86.90**	**29.68**	**2.58**	**75.10**	**20.45**	**2.23**	**78.30**	**4.40**	**1.44**	**97.50**	**29.85**	**3.20**	**74.30**

Table 1. **Rotation estimation evaluation on the InteriorNet, the SUN360, and the StreetLearn datasets.** We report the mean and median geodesic error in degrees, and the percentage of pairs with a relative rotation error under 10°, for different overlapping levels (Large, Small, and None), as detailed in Section 4.3 of the main paper. For the percentage of pairs (10°%), higher is better. Models trained only on overlapping pairs are denoted with "-o". *Errors are computed only over successful image pairs, for which these algorithms output an estimated rotation matrix (failure over more than %50 of the test pairs is shown in gray).

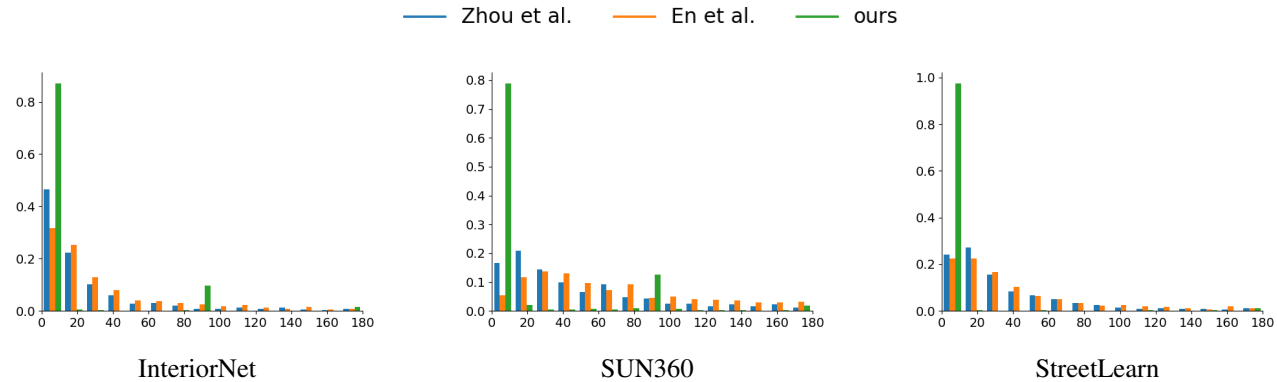
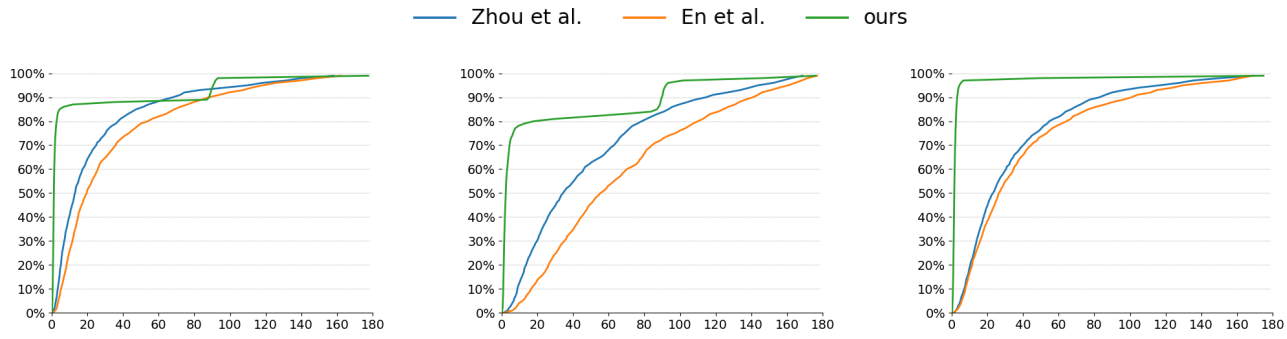


Figure 2. Geodesic error histograms. The x-axes is the geodesic error (each bin covers 10°, *e.g.* the first bin is over errors in the range $[0°, 10°]$, and so on). The y-axes is the frequency.

%Overlap	Method	InteriorNet			InteriorNet-T			SUN360			StreetLearn			StreetLearn-T		
		Success	Fail	Total	Success	Fail	Total	Success	Fail	Total	Success	Fail	Total	Success	Fail	Total
Large	SIFT [8]	371	32	403	252	83	335	201	2	203	166	4	170	50	102	152
	D2-Net [4]	330	73	403	102	233	335	176	27	203	119	51	170	3	149	152
	SuperPoint [3]	382	21	403	259	76	335	201	2	203	169	1	170	30	122	152
Small	SIFT [8]	195	97	292	121	203	324	217	49	266	267	34	301	39	278	317
	D2-Net [4]	17	275	292	0	324	324	23	243	266	17	284	301	1	316	317
	SuperPoint [3]	112	180	292	60	264	324	112	154	266	115	186	301	3	314	317
None	SIFT [8]	8	297	305	5	336	341	32	499	531	27	502	529	15	516	531
	D2-Net [4]	0	305	305	0	341	341	2	529	531	0	529	529	0	531	531
	SuperPoint [3]	1	304	305	0	341	341	4	527	531	0	529	529	0	531	531
All	SIFT [8]	574	426	1000	378	622	1000	450	550	1000	460	540	1000	104	896	1000
	D2-Net [4]	347	653	1000	102	898	1000	201	799	1000	136	864	1000	4	996	1000
	SuperPoint [3]	495	505	1000	319	681	1000	317	683	1000	284	716	1000	33	967	1000

Table 2. **Number of pairs for which SIFT, D2-Net, and SuperPoint return an answer.** For each dataset and overlap ratio, we report the number of pairs for which RANSAC successfully outputs a model (regardless of whether that model is accurate or not).



InteriorNet SUN360 StreetLearn

Figure 3. Cumulative distribution of geodesic error. The x-axes is the geodesic error, and the y-axes is the percentage.

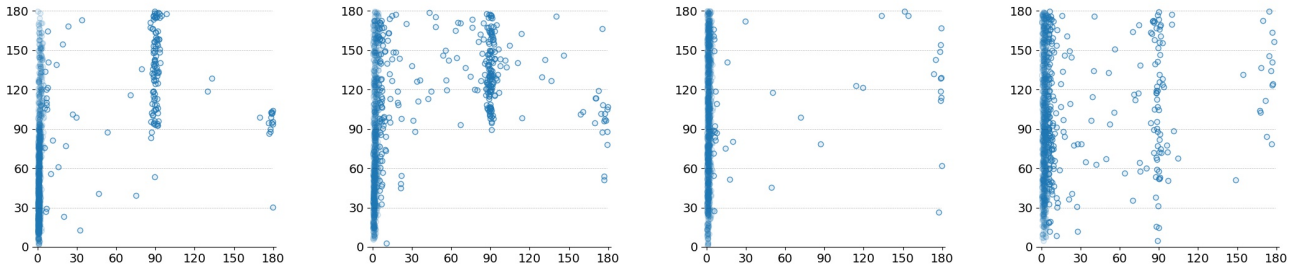
errors smaller than 10° ($\%99.7$). This result suggests that pitch can be predicted from a single image. We should note that pitch angles are generally easier to reason about (as can be seen in the ablation study in the main paper). This experiment also shows that the model correctly predicts an identity rotation matrix (when the same image is fed to the network twice), with an average geodesic error of 0.62° over the StreetLearn dataset.

3. Additional Qualitative Results

We visualize some failure cases in Figure 5. On the StreetLearn dataset, the error is mainly derived from the ambiguity of opposite street directions. For example, the top two rows show a street crossing. In both cases, the model correctly predicts the pitch angle, but is confused in predicting the correct direction at the intersection. The remaining two rows show that when the image pair—while non-overlapping—has similar objects visible in the view (e.g., green scaffolding sheds in the last row), the model might regard the two images are pointing in the same direction, and when the image pair only shares has a small overlap, as in the second to last row, the model may predict it as non-overlapping pairs, and both of these cases have 180° error.

On SUN360 and InteriorNet, the larger variety of indoor scene configurations makes the relative rotation task more difficult. For example, the first two rows contain repetitive texture of the roof and floor, which confuse the model. The last two rows illustrate ambiguities arising from non-overlapping pairs, where the possible horizontal rotation range is $[90^\circ, 270^\circ]$. Even when we narrow down the choices, for example if both views observe a corner of the room, there still might exist three plausible choices. Our results suggest that estimating rotations in indoor scenes is generally more difficult in comparison to outdoor scenes.

In Figure 6, we provide additional visualizations of cues detected in image pairs. We provide more qualitative results on StreetLearn, SUN360, and InteriorNet in Figure 7 and Figure 8. More qualitative results on London and Pittsburgh are shown in Figure 9.



InteriorNet SUN360 StreetLearn StreetLearn-T

Figure 4. Geodesic error distribution. The x-axes is the geodesic error, and the y-axes is the ground truth rotation angle.

		Rotation Error			Yaw Error			Pitch Error		
%Overlap	Roll	Avg(°↓)	Med(°↓)	10°(%↑)	Avg(°↓)	Med(°↓)	10°(%↑)	Avg(°↓)	Med(°↓)	10°(%↑)
Large	$= 0°$	2.27	1.07	99.41	1.86	0.67	99.41	0.70	0.53	100.00
	$< 5°$	7.71	3.18	97.08	5.20	0.84	97.65	1.55	1.55	99.41
Small	$= 0°$	2.66	1.41	98.34	2.12	0.87	98.67	0.87	0.61	99.67
	$< 5°$	6.43	3.39	96.33	3.43	0.95	98.01	1.56	0.87	99.00
None	$= 0°$	6.48	1.58	96.60	5.70	0.92	96.79	1.18	0.59	99.24
	$< 5°$	8.54	3.44	96.22	6.88	1.16	96.22	1.37	0.87	99.24
All	$= 0°$	4.61	1.42	97.60	3.97	0.85	97.80	1.00	0.59	99.50
	$< 5°$	7.77	3.39	96.40	5.56	1.04	97.00	1.46	0.88	99.20

Table 3. **Roll angle experiments on StreetLearn**, evaluating the effect of adding small roll angles to image pairs at test time.

References

[1] Matthew Brown, Richard I Hartley, and David Nistér. Minimal solutions for panoramic stitching. In *CVPR*, pages 1–8. IEEE, 2007.

[2] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. *arXiv preprint arXiv:1606.03558*, 2016.

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018.

[4] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR*, 2019.

[5] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In *ECCV*, pages 0–0, 2018.

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM (CACM)*, 24(6):381–395, 1981.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016.

[8] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[9] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, pages 1874–1883, 2016.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[12] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, pages 5745–5753, 2019.

Figure 5. **Failure cases.** The full panoramas are shown on the left, with the ground-truth perspective images marked in red. We show our predicted viewpoints in yellow.

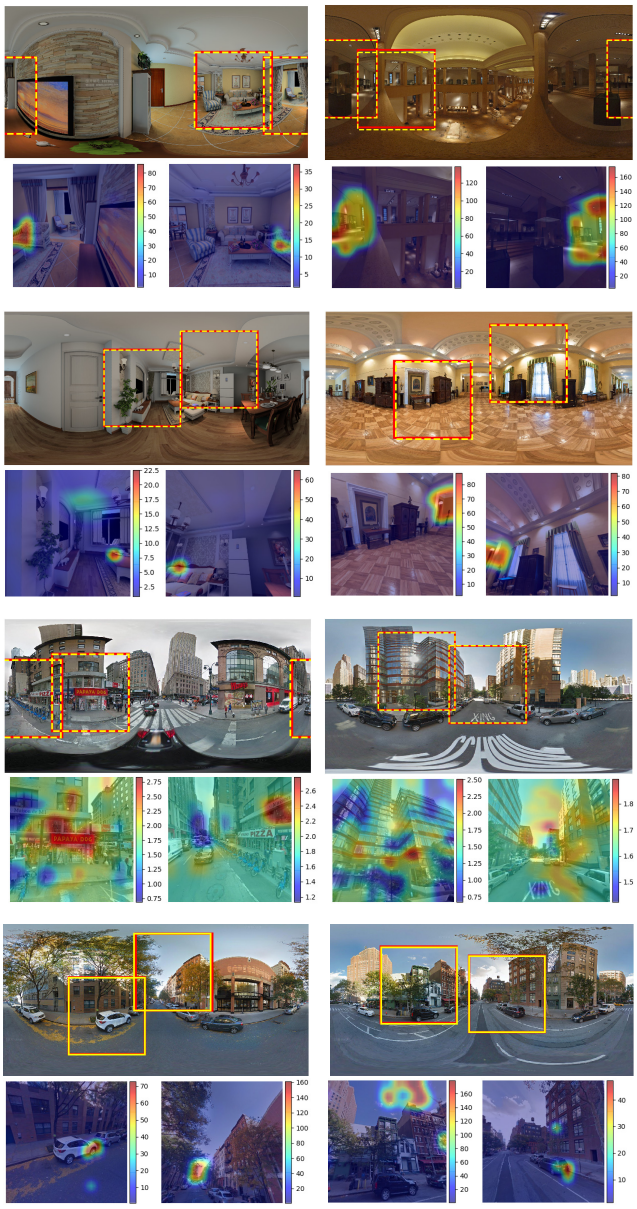


Figure 6. **Visualizing cues detected by our model for overlapping and non-overlapping pairs.** We show regions which, when blocked, affect the rotation error, with warmer colors depicting larger errors (according to their associated color bars). The full panoramas are shown above, with the ground-truth and predicted perspective image regions marked in red and yellow, respectively. In indoor scenes, blocking corresponding regions (in overlapping pairs) may result in large errors (first row). In outdoor scenes that are richer in information, blocking small regions does not seem to affect the model's predictions in overlapping cases (third row). In non-overlapping cases, we can see that the model seems to reason about truncated objects (second row, left) or cues related to vanishing points (bottom row, left), sunlight (bottom row, right) and shadows (second row, right).

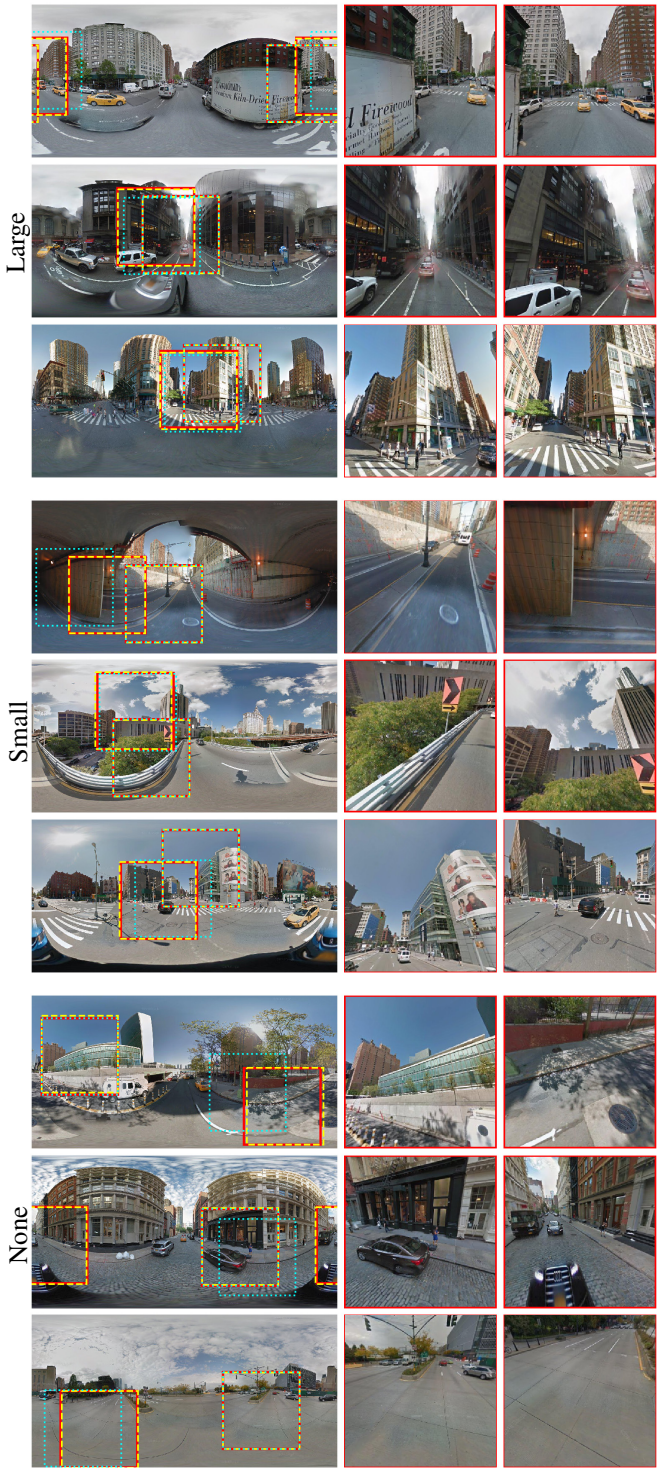


Figure 7. **Predicted rotation results.** Full panoramas are shown on the left, with the ground-truth perspective images marked in red. We show our predicted viewpoints in yellow, and results obtained using the regression model of Zhou *et al*. [12] in blue.

Figure 8. **Predicted rotation results of indoor datasets.** Full panoramas are shown on the left, with the ground-truth perspective images marked in red. We show our predicted viewpoints in yellow, and results obtained using the regression model of Zhou *et al*. [12] in blue.

Figure 9. **Predicted rotation results on new cities.** Full panoramas are shown on the left, with ground-truth perspective images marked in red. We show our predicted viewpoints in yellow and results obtained using the regression model of Zhou *et al*. [12] in blue.