Learning Camera Localization via Dense Scene Matching Supplementary Material

Shitao Tang[1] Chengzhou Tang[1] Rui Huang[2] Siyu Zhu[2] Ping Tan[1]
[1]Simon Fraser University [2]Alibaba A.I Labs
{shitao_tang, chengzhou_tang, pingtan}@sfu.ca {rui.hr, siting.zsy}@alibaba-inc.com

This Supplementary provides the following contents: 1) The architecture of Net_{coords} and Net_{conf} as described in Sec.3.3.2 and Sec.3.3.3 of the main paper. 2) Additional analysis on the effectiveness of top K sorting strategy and the selection of parameters, i.e., image resolution and the number of scene images. In the end, we present more visualization results on the comparison of estimated coordinate maps with different methods.

1. Archtecture of Net_{coords} and Net_{conf}.

Fig. 1 shows the architecture of Net_{conf} and Net_{coords}. The input of Net_{coods} is a $H^l \times W^l \times 4K$ ($K = 16$ in implementation) cost-coordinate volume formed by concatenating the cost volume with 3D scene coordinates. As shown in Fig. 1, Net_{coods} consists of 3 residual blocks [1] and one denseblock [2]. The residual blocks consist of 1×1 convolutional layer. It takes input of cost-coordinate volume and generates a $H^l \times W^l \times 64$ coordinate feature map. Then, the scene coordinate map is estimated by the denseblock, which takes the concatenation of image features, coordinate features and the initial coordinate map up-sampled from last layer (if applicable). On the other hand, Net_{conf} consists of 5 residual block with context normalization [3]. It takes the concatenation of the estimated scene coordinate map with the corresponding 2D pixel coordinate map and estimates a confidence score for each pixel.

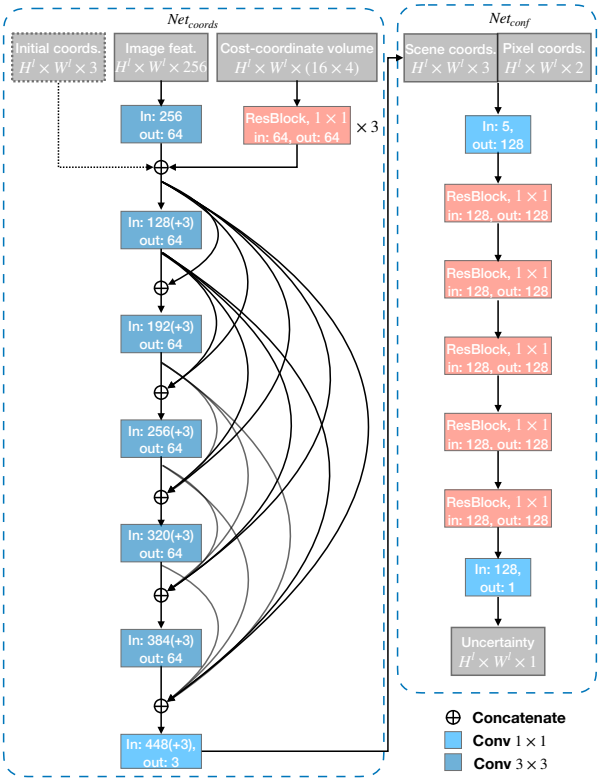


Figure 1: Archtecture of Net_{conf} and Net_{coords}. We use residual block for Net_{conf} and dense block for Net_{coords}

2. Additional Analysis

This section provides additional analysis of DSM. All the experiments are conducted on 7scenes dataset. The data processing and training process are the same as described in the main paper. At the inference time, We use 1 out of every 10 frames for each sequence. Pose accuracy, the percentage of predicted poses falling within the threshold (5°, 5cm), is used as the evaluation metric.

Effects of correlation sorting. As described in Sec.3.3.1 of the main paper, one of the procedures in cost volume construction is sorting and selecting top K coordinates for each pixel from the correlation tensor. The motivation behind this operation is two-fold. Firstly, as the number of retrieved scene images varies, top K selection results in a cost volume with a fixed size. Secondly, a sorted cost volume leads to a more accurate estimated coordinate map. To verify the effectiveness of correlation sorting, we fix the scene image number to 5 and directly use the correlation tensor as the cost volume for coordinate map regression. The results are shown in Table. 1. It can be seen that the estimated pose accuracy improves by correlation sorting consistently on all sequences. Moreover, since top K sorting and selection re-

	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
No sorting	0.82	0.74	0.85	0.72	0.43	0.58	0.05
Sorting	0.96	0.95	1.0	0.88	0.53	0.72	0.66

Table 1: Pose accuracy with/without top K correlation sorting. The estimated pose accuracy improves by correlation sorting consistently on all sequences.

Num.	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
1	0.87	0.85	0.87	0.71	0.45	0.63	0.17
3	0.90	0.94	0.91	0.79	0.46	0.67	0.20
5	0.94	0.94	0.94	0.80	0.54	0.68	0.24
10 ($\star$)	0.96	0.95	1.0	0.88	0.53	0.72	0.66

Table 2: Pose accuracy with respect to the number of scene images. The network is trained and tested with the corresponding number of scene images except the one with 10 scene images. The notation ($\star$) means we train the network with 5 scene images instead of 10 scene images.

Reso.	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
192×256	0.92	0.84	0.89	0.78	0.49	0.64	0.23
384×512	0.96	0.95	1.0	0.88	0.53	0.72	0.66

Table 3: Pose accuracy with respect to different image resolutions. In our implementation, We resize all images to resolution of 384×512 for better efficiency and performance.

sults in a fixed-size cost volume, we can use different scene image numbers for training and testing. During the training process, the scene image number can be fixed for better efficiency while for inference we can leverage more scene images for higher accuracy.

Number of scene image. To show the effects of scene image number N, we change N from 1 to 10 in the training and testing process to evaluate the pose accuracy. The model is re-trained with respect to the corresponding scene image number for $N = 1, 3, 5$. Since training with more than 5 scene images leads to unacceptable GPU memory consumption, we still use 5 scene images in training when testing with 10 scene images. As shown in Table 2, increasing N from 1 to 5 results in higher pose accuracy. In addition, we can see that 10 scene images obtain higher performance than 5 scene images. This indicates that even if the model is trained with fewer scene images, leveraging more scene images leads to better performance. Considering the trade-off between performance and efficiency, we set $N = 10$ in the main paper.

Image resolution. We test our model using 2 different image resolution size 192×256 and 384×512. As shown in Table. 3, we can see the resolution of 384×512 outperforms 192×256. A higher resolution than 384×512 could consume more GPU memory and lead to slower running time. Therefore, we resize all the images to 384×512 in our system for better efficiency.

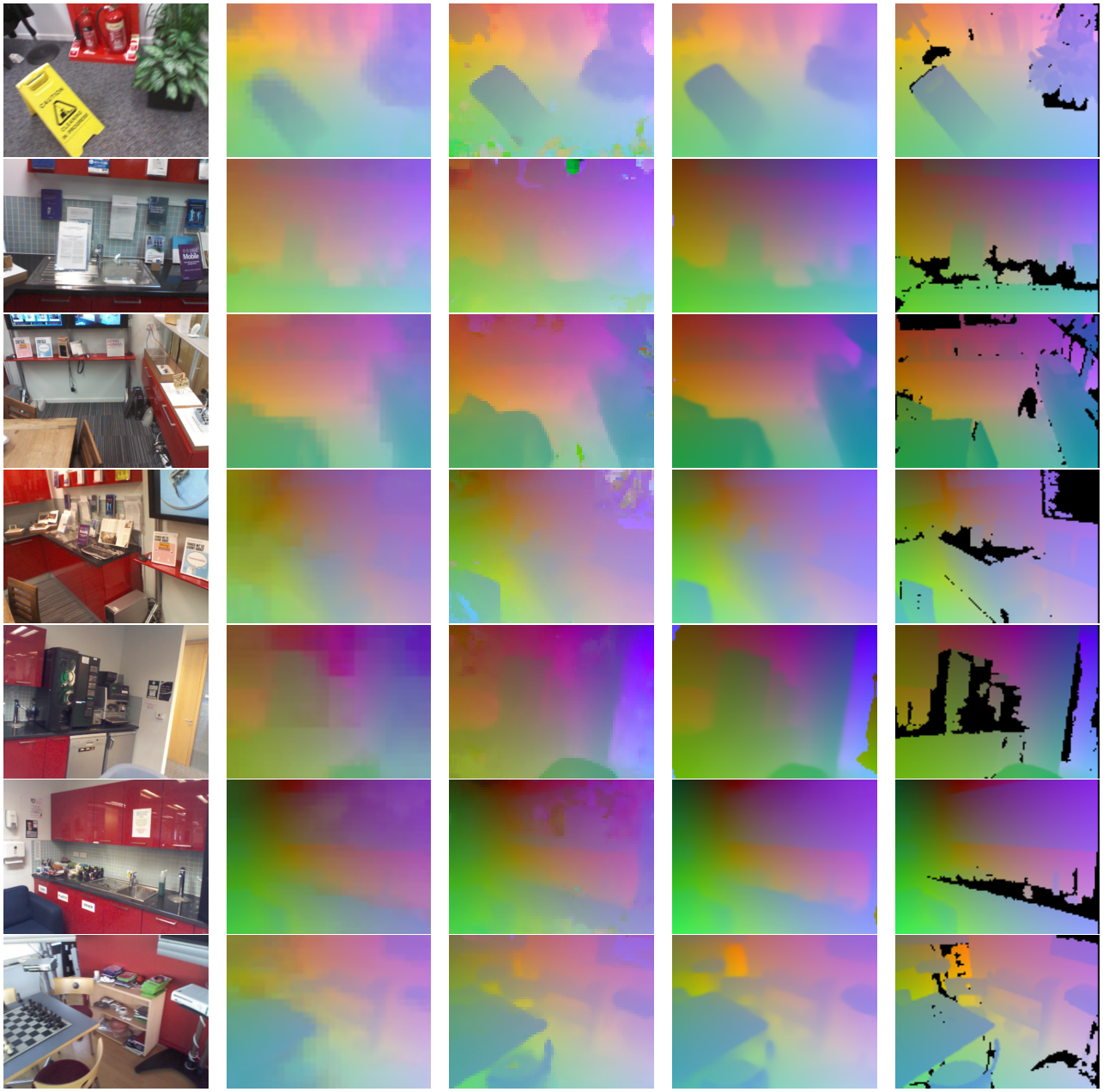
More coordinate map visualization. Fig. 2 provides more visualization results on the comparison of estimated coordinate maps from SANet, DASC++, and our method (DSM). In general, the coordinate maps produced by DSM recover more details as in the ground truth and have fewer artifacts than SANet and DSAC++.

References

[1] 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

[2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1

[3] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2666–2674, 2018. 1



(a) Query (b) SANet (c) DSAC++ (d) DSM (e) G.T.

Figure 2: Coordinate map visualization for SANet, DSAC++ and DSM. The coordinate maps produced by DSM recover more details as in the ground truth and have fewer artifacts than SANet and DSAC++.