PATS: Patch Area Transportation with Subdivision for Local Feature Matching – Supplementary Material –

Junjie Ni[1,2]*, Yijin Li[1,2]*, Zhaoyang Huang[3], Hongsheng Li[3], Hujun Bao[1,2], Zhaopeng Cui[1] and Guofeng Zhang[1,2]†
[1]State Key Lab of CAD&CG, Zhejiang University [2]ZJU-SenseTime Joint Lab of 3D Vision
[3]Multimedia Laboratory, The Chinese University of Hong Kong

In this supplementary document, we provide additional experiments in Sec. A, describe more implementation details in Sec. B, discuss the computational overhead in Sec. C and discuss the limitation in Sec. D. Finally, we show more qualitative results in Sec. E. In this document, references that point to the main manuscript will be referenced as "P-".

A. More Experiment Results

A.1. Qualitative Comparison on Self-captured Data

To verify the generalization performance of our method, we recorded several video sequences that capture the scene from near to far. We show the results in Fig. A where only accurate matches whose epipolar error is less than 0.5 pixel are drawn. To compute the epipolar error, we first recover the camera pose of the video sequence through the technique of structure from motion [9]. It can be seen that our method stably obtains dense and accurate matches. On the contrary, other methods struggle to achieve sufficient matches when the image pair suffer from large-scale variant. More qualitative results can be found in our supplementary video.

A.2. Two-view Reconstruction

Since our method achieves high-precision matches that are densely and uniformly distributed in the images, we can obtain semi-dense reconstruction by simply triangulating the matches in a image pair. As shown in Fig. B, although the image pair suffer from viewpoint, scale ,and illumination variant, our method enables us to recover semi-dense point cloud that is high-quality. More qualitative results can be found in our supplementary video.

A.3. Evaluation of Inlier Coverage on MegaDepth

Apart from matching precision and matching coverage shown in Table P-3, here we evaluate a new metric that both consider the accuracy and the distribution of the matches. The metric is inlier coverage. Compared with matching coverage, the inlier coverage only considers the distribution of accurate matches where a match is regarded as accurate if its epipolar error is less than a threshold. In Fig. C, we show the mean inlier coverage across all resolutions within different error thresholds. It can be seen that our method achieves much higher inlier coverage within all error threshold.

A.4. Visualization of the Area Transportation

As shown in Fig. D, PATS can detect the related region straightforwardly for patches in the source images and compute accurate areas for them. Therefore, we can generate correct new image pairs after resizing the associated regions to the same scale and then match them in more detail in the following iterations, which reduces a big and hard matching problem to many small and simple ones.

A.5. Failure Cases

Even though PATS is much more robust than other methods in scale variation, it may still fails when encountering extreme viewpoint change. In Fig. E, we visualize two failure examples for PATS on the ScanNet dataset where the image pairs are under extreme viewpoint change and have very limited overlapping regions.

B. Implementation Details

B.1. Architecture Details

Feature Extraction. In the first layer, we extract 512-dimensional features at each of the three pyramid levels in images, then reduce the feature dimensions to 256, 128, and 64 dimensions through MLP networks, and concatenate them, resulting features with 448 dimensions. In the second level, we reduce the 448-dimensional features from the first

*Junjie Ni and Yijin Li contributed equally to this work.
†Guofeng Zhang is the corresponding author.

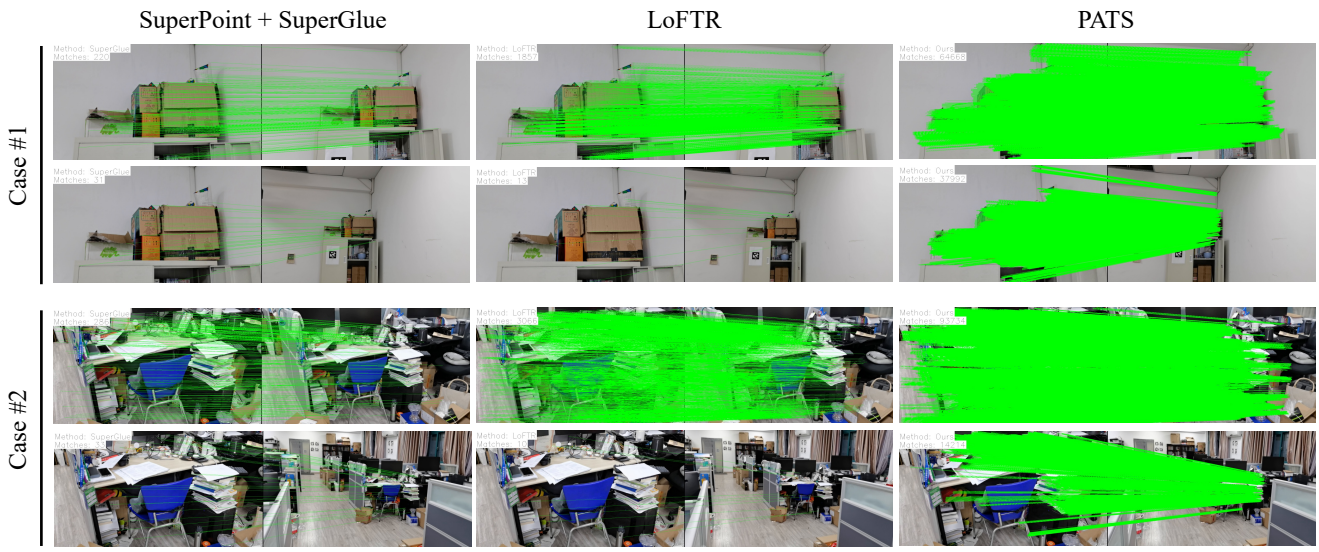

Figure A. Qualitative comparison on our self-captured data. Our method stably obtains dense and accurate matches. On the contrary, other methods are struggle to achieve sufficient matches when the image pair suffer from large scale variant.

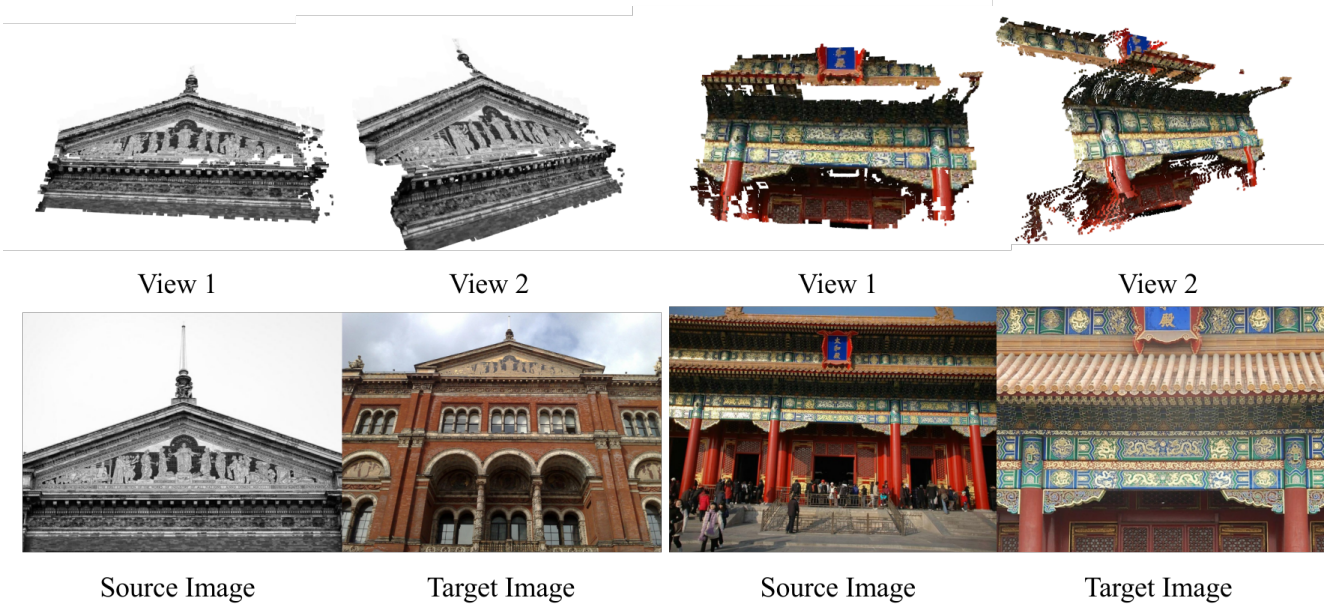

Figure B. The semi-dense point cloud reconstructed through two-view triangulation. Although the image pair suffer from viewpoint, scale and illumination variants, our method are capable of recovering semi-dense point cloud that are high-quality.

level to 8 dimensions by an MLP network and then concatenate them with the new 256-dimensional features, resulting 264-dimensional features. In the third layer, considering insufficient local information in this layer, we further add the output of each convolutional layer from the second layer with the corresponding third-layer features to ensure adequate encoding of information and finally obtains 128-dimensional features. The number of transformer layers is set as 9, 9, and 5 for each of the three iterations.

Variants of Concentration Loss. During training, we employ a variant of concentration loss, which is defined as:

$$L_c = \frac{1}{|\mathcal{M}_i|} \sum_{(i,j)\in\mathcal{M}_i, j'\notin B_i} (P_{i,j'} + P_{M+1,j'}) \qquad (1)$$

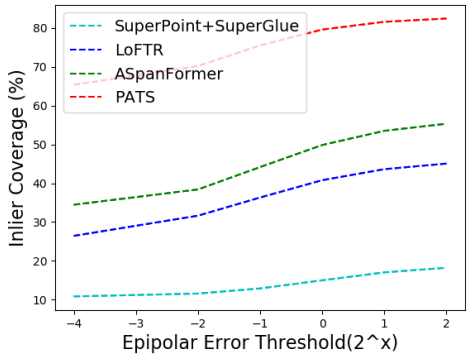

Figure C. Evaluation of inlier coverage on the MegaDepth dataset. Our method achieves much more higher inlier coverage than other methods within all error threshold.

where line $M+1$ of P refers to the rubbish bin, and $P_{M+1,j'}$ means that we want the area of every patch in the bounding box not to be transported to the trash. Empirically, we found that this slight modification leads to faster convergence.

Setting in Different Layers. In our experiment, we do not conduct image resizing and patch trimming after the second layer considering the efficiency of our implementation. Furthermore, we do not conduct flood fill when computing the bounding box in the third layer. Instead, we fix it with the size of 5×5.

B.2. Training Data

MegaDepth. Based on the training split from SuperGlue [8], we select the image pairs from the MegaDepth dataset if they co-view more than 50 map points. Then, with the ground truth depth map, we divide the source image into 32×32 patches and re-project each center point of the patches into the target images. We consider it as a positive sample if the re-projection points have valid ground truth depth. We regard it as a negative sample if the re-projection fall outside the target image. Finally, we select those image pairs that include more than 20 positive samples and more than 1 negative sample as our training set.

ScanNet. Based on the training split from SuperGlue [8], we select the image pairs from the video sequences with an fixed interval. The pairs are selected with an interval of 100 frames for training the first layer and with an interval of 50 frames for training the rest layers.

B.3. Training Details

In order to adapt our model to multiple resolutions and to converge faster, we choose to train our first layer of the network on 640×480 resolution images with a batch size of 128 at first and then fine-tune on 1600 long-edge resized image pairs with a batch size of 4. We train the second and third layers on 640×480 resolution images. Furthermore, we utilized 3 RTX3090s to train our model, and fine-tuned it on 1600 long-edge resized image pairs using 4 RTX3090s

B.4. Experiment Details

Pose estimation. We set the RANSAC threshold for pose estimation in outdoor cases as 0.25 pixel, while in indoor cases we set it to 0.4 pixel. We pad the images to make them divisible by the patch size in our first layer, which is 32×32.

Building Up Extreme-scale Dataset. We sample 1000 image pairs from the MegaDepth dataset. The source images are resized to 640×480, and we manually scale the target image to five different resolutions from 320 to 1600 along the longer side, which make up our extreme-scale dataset. To make the resolution of the image pair equal-sized, we fill the smaller image with an unrelated background image that has been resized to equal the size of the other image in the image pair. We show two sample pairs and corresponding matching results in Fig. F.

Visual Localization on the Aachen day-night Dataset. We use the visual localization toolbox HLoc [7] for pose estimation on the Aachen day-night dataset [12]. HLoc follows the common Structure-from-Motion (SfM) pipeline. It first reconstructs a 3D model using the local features and matches and then consecutively registers the repeatable keypoints in the new image into the model. However, our method and other detector-free methods [1, 10] directly regress matches from a pair of images and thus do not obtain repeatable keypoints across multiple images. Specifically, To address the problem, previous works [10, 13] either takes average among a patch or takes the most plausible point positions as the keypoints. Inspired by them, we quantize our matches by their weighted average positions to represent key points close to 4 pixels from each other, which means sacrificing pixel-level accuracy here, and with more observations, we compute the average position on more points and therefore sacrifice more pixel-level precision. Therefore, we get much better localization results on the night track of Aachen day-night with 12984 database image pairs, while we get slightly worse localization results on the full track of Aachen day-night v1.1 with 133066 database image pairs. After quantization, we only keep one point in an 8×8 gird, thus removing duplicate matches. Although this is not a perfect solution, it is better than only using point observations from source images.

B.5. Definition of Evaluation Metrics

Here we explain the matching precision and matching coverage metrics for evaluation on the proposed extreme-scale dataset. Matching precision measures the ratio of accurate matches where a match is defined as accurate if its epiploar error is less than 0.5 pixel. The matching cover-

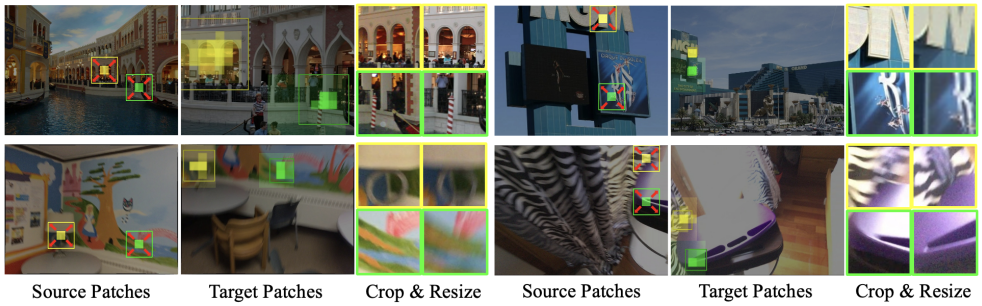

Figure D. **The Visualization of Patch Area Transportation.** After patch area transportation, we find the corresponding position in the target image $\hat{\mathbf{p}}_i$ for each source patch i, and then determine the window size to be cropped. The window size in the source image is fixed by e and the window size in the target image is computed from transportation matrix $\mathbf{P}$, which ensures the target window covers the source window's content. After resizing cropped windows, the image content is aligned to the same scale, which eases the feature extraction. The color in the target images indicates the transportation area where the bright color represents the big transportation area and the dark color represents the small transportation area.

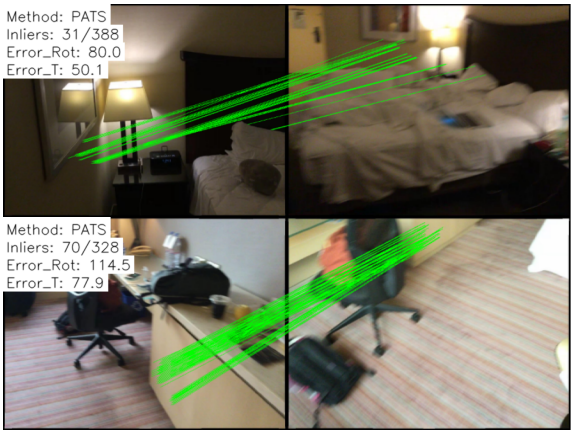

Figure E. **Failure Cases.** We visualize two failure cases for PATS on the ScanNet dataset where the image pairs are under extreme viewpoint changes.

age, on the other hand, validates the distribution of matching, which is introduced in colmap [9]. Given an image pair, we re-project the pixels from the source image to the target image via the camera pose and its depth to obtain the co-visible area between the image pair. Then, we partition the area into a grid with equal-sized cells of 8×8. Each cell is labeled full if an accurate match locates within it, otherwise, it is empty. The matching coverage is computed as the full cell rate. For the AUC metrics used on the pose estimation, please refer to SuperGlue [8]. For the AEPE and Fl-all metrics used on the optical flow, please refer to COTR [5].

C. Computational Overhead

Running on NVIDIA RTX 3090, PATS (900ms) is slower than LoFTR (64ms) but is comparable with PDC-Net+ (850ms) when processing 640×480 images. Besides, PATS (5.0GB) occupies more memory than LoFTR (3.6GB) but less memory than PDC-Net+(5.6GB). However, thanks to the scale-adaptive subdivision, for 1600×1200 images, PATS (8.3GB) consumes less memory than LoFTR (22GB) and PDC-Net+(23GB).

D. Limitations

Our current system has some limitations, which could be fixed in future works. First, in this work, we focus on solving scale problems through our patch area transportation and subdivision. Currently, we do not consider the matching challenge brought by other geometric deformations like rotation. It would be an interesting topic to integrate these geometric transformations (e.g., rotation, homography, and affine) into our transportation-based framework. Another problem is that our method is not fast enough. It takes about 900ms to process a 640 ×480 color image on a single GPU (NVIDIA RTX 3090), which is acceptable for downstream applications like image editing and structure from motion(SFM) but might be insufficient for SLAM. Note that, in this work, we did not optimize the network architecture for run-time efficiency, and many standard approaches, such as [3, 4, 6], can be directly adopted. Lastly, removing the regular square shape of the patch may further improve the quality, and we will explore deformable patches in future work.

E. More Visualizations

We visualize more qualitative results on YFCC100M [11] and Scannet [2] dataset in Fig. G and Fig. H. Compared with other methods, PATS achieves dense and accurate feature matching, which both contribute to high-precision pose estimation and visual localization.

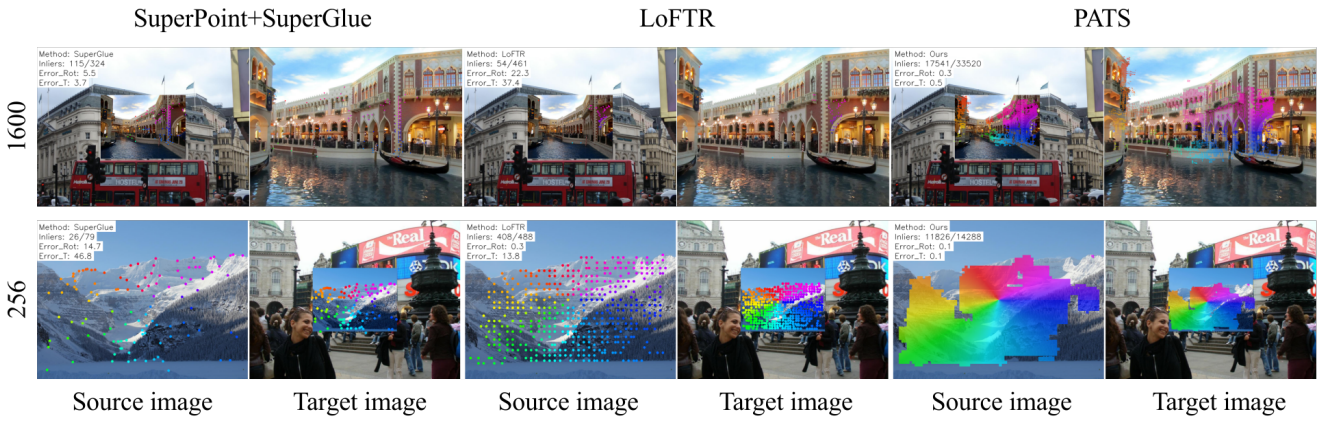

Figure F. Two sample pairs and corresponding matching results on the extreme-scale dataset. The source images are resized to 640×480 in all cases. The target images are scaled to different resolutions from 256 to 1600 along the longer side. We show two cases where the target image have the resolutions of 1600 and 256, respectively. To make the resolution of the image pair equal-sized, we fill the smaller image with an unrelated background image which have already been resized to be equal-sized with the larger one.

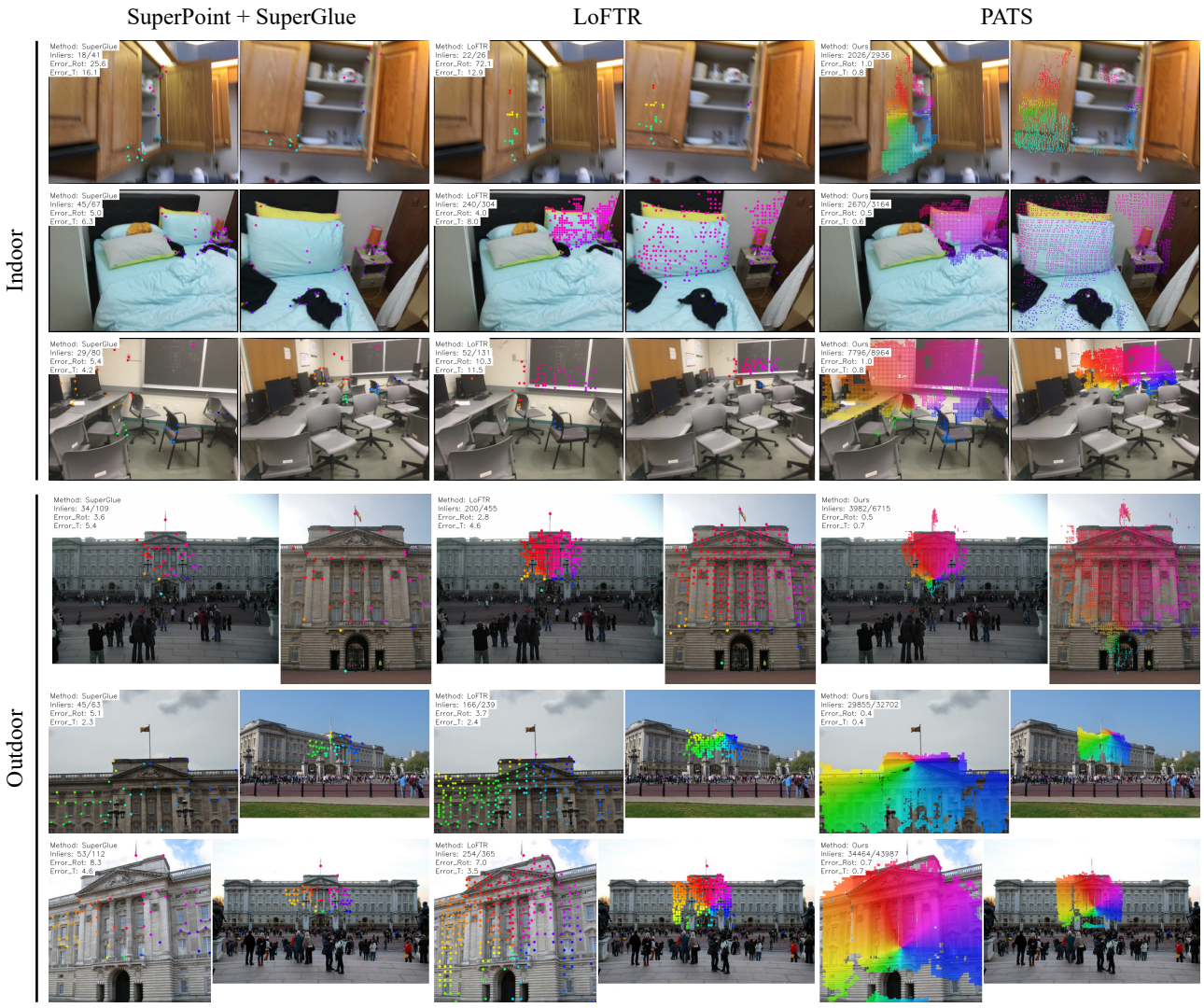

Figure G. More qualitative comparison on Scannet and YFCC100M dataset. These images show our superiority over accuracy and coverage, which is why our method outperforms previous methods in pose estimation. The matched features are visualized as the same color. We have filtered incorrect matches that have large epipolar error.

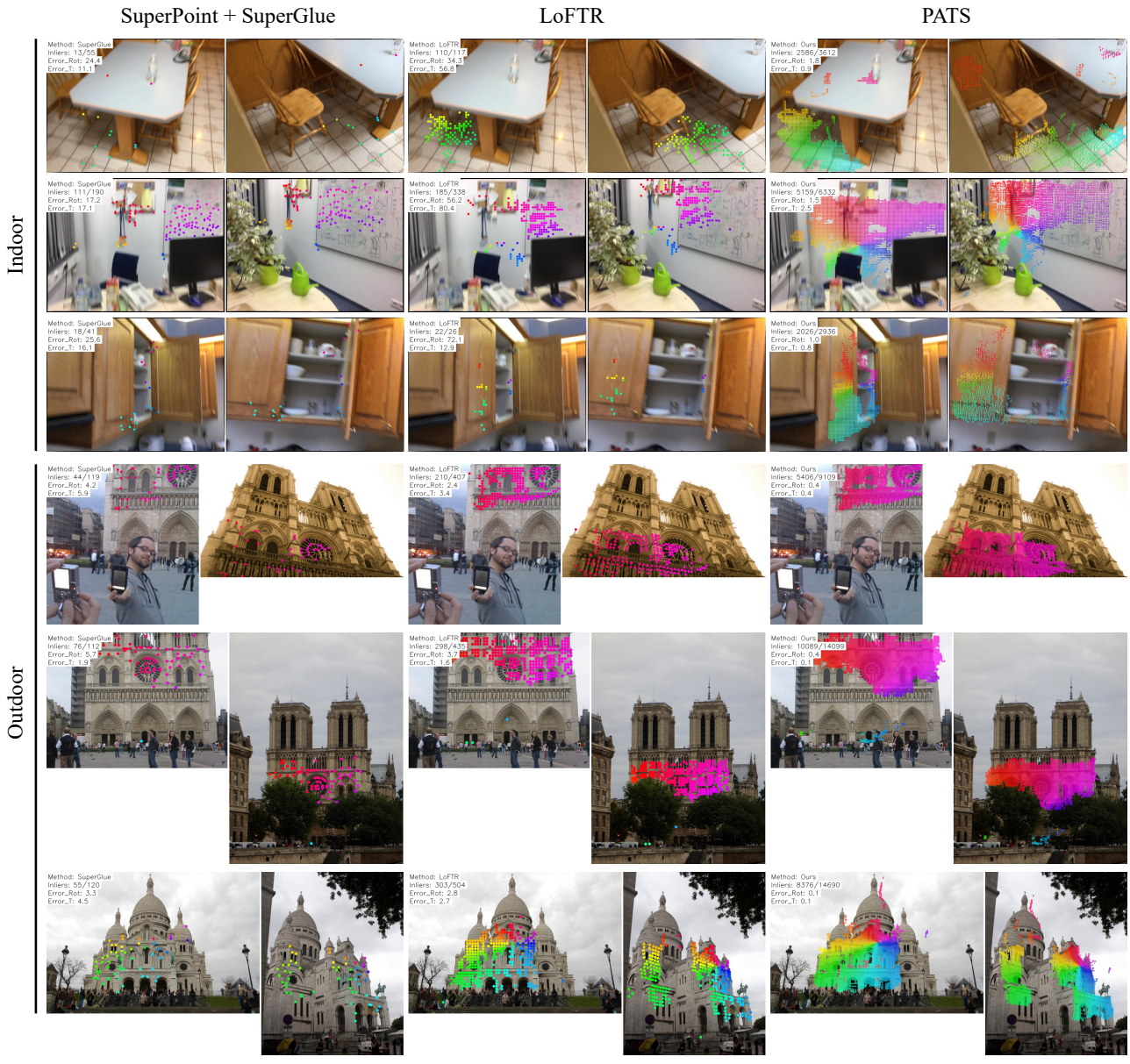

Figure H. More qualitative comparison on Scannet and YFCC100M dataset. These images show our superiority over accuracy and coverage, which is why our method outperforms previous methods in pose estimation. The matched features are visualized as the same color. We have filtered incorrect matches that have large epipolar error.

References

[1] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *European Conference on Computer Vision*, 2022. 3

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 4

[3] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021. 4

[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 4

[5] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. COTR: correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 171–180. Springer, 2021. 4

[6] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *Pattern Recognit.*, 105:107281, 2020. 4

[7] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 3

[8] 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. 3, 4

[9] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1, 4

[10] 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. 3

[11] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: the new data in multimedia research. *Commun. ACM*, 59(2):64–73, 2016. 4

[12] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *Int. J. Comput. Vis.*, 129(4):821–844, 2021. 3

[13] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixé. Patch2pix: Epipolar-guided pixel-level correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4669–4678, 2021. 3