Supplementary Materials
3DFA: Aligning the Features Between Point Cloud and Query Image for Scene-Specific Visual Localization

Anonymous CVPR submission

Paper ID ****

1. Implementation Details

1.1. Trainable Parameters

The trainable parameters of 3DFA include

- The feature head of R2D2 network [2].
- ϕ_1, the 2D patch encoder.
- ϕ_2, the 3D patch encoder.
- ϕ_3, the 2D-3D fusion layer.
- ϕ_4, the 2D patch decoder.
- ϕ_5, the 3D patch decoder.
- The cross attention layer in the final pose estimation module.
- ϕ_6, the reliability score layer.

Among all these trainable parameters, ϕ_1 to ϕ_6 are implemented as MLPs, where ϕ_1 to ϕ_5 uses ReLU activation and ϕ_6 uses Sigmoid activation. The cross attention layer as well as the 3D positional encoding is borrowed from the widely used point transformer [8]. Note that the keypoint extraction head of R2D2 network is freezed, as we only want to finetune the feature extraction part.

1.2. Hyperparameters

In summary, the hyperparameters of 3DFA include

- α, β, the loss factor.
- M, the final pose estimation scope.
- S, the reliability score threshold.

Our experimental results in the paper (Table 1∼3 in main paper) are obtained with $\alpha = \beta = 1$,$M = 5$, and $S = 0.7$. The experiment on α and β is included in later part.

1.3. Training and Inference

3DFA is trained on RTX4090 GPU. In DL3DV-10K dataset, training takes around 22 hours for one scene. This period does not include the point cloud reconstruction and 2D keypoint extraction, which we consider as data preprocessing. During inference, 3DFA is able to process 0.54 image per second. One more advantage of 3DFA is that, since it replaces support images by the point cloud for feature alignment, our approach does not require support images during inference. This makes it easier to deploy 3DFA to real applications, as we only need the trained weights and the point cloud.

2. Data Preparation Details

3DFA employs PNeRF [9] as the 3D reconstruction approach which also requires depth maps as point cloud initialization. ***7Scenes*** [4] already provides ground truth depth maps. For ***DL3DV-10K*** and ***ScanNet++***, we use COLMAP [3] to estimate the depth maps. If a scene is named by hashes, we assign it a casual name. We follow previous work [10] for the real world scale information and apply a rough scale when the ground truth is missing.

3. More Experimental Results

3.1. Comparison Schemes Setup

To have a comprehensive evaluation of model performance, we include both scene-agnostic and scene-specific schemes in our experiments. However, since we focus on scene-specific visual localization in this paper, we ensure a fair comparison by feeding all networks the same set of training data. All results for one scene is obtained on that specific scene using the same set of support and query images. It is infeasible to train those foundation models again [1, 5–7], so for them we only run inference. We find that 16 images are enough to yield their maximum capabilities roughly, so for each query image, we feed the 16 most similar support images to the foundation models.

3.2. Positions of Patch-Matching Loss

As mentioned in the paper, an ascending number of patches result in the best localization performance. Our final 3DFA uses a "$3, 5, 7, 9$" number of patches. Here we also explore the positions of patch-matching loss inside these different 3DFA layers. The results in Table 1 indicate that applying the patch-matching loss at every layer is essential, as

Table 1. Median translation error (centimeters) on the ScanNet++ dataset of 3DFA when patch-matching loss is removed in a certain layer. For example, "w/o $\mathcal{L}_p$ @3" means to remove patch-matching loss in the first 3DFA layer.

3DFA	Work	Studio	Bedroom	Living
w/o $\mathcal{L}_p$ @3	5.0	5.3	6.0	4.8
w/o $\mathcal{L}_p$ @5	5.1	5.3	5.9	4.8
w/o $\mathcal{L}_p$ @7	5.1	5.4	5.9	4.8
w/o $\mathcal{L}_p$ @9	5.2	5.5	6.0	**4.7**
w/ all $\mathcal{L}_p$	**4.8**	**5.2**	**5.8**	**4.7**

Table 2. Median translation error (centimeters) on the ScanNet++ dataset of 3DFA with differemt α (factor of patch-matching loss).

α	Work	Studio	Bedroom	Living
0	5.4	5.5	6.0	4.8
0.25	5.3	5.5	5.9	4.8
0.5	5.0	5.3	5.9	4.8
0.75	**4.8**	**5.2**	**5.8**	4.8
1	**4.8**	**5.2**	**5.8**	**4.7**
1.25	4.9	**5.2**	**5.8**	4.8
1.5	5.1	5.3	6.0	5.1
2	5.3	5.5	6.1	5.2

removing it from any layer leads to a measurable drop in performance. This observation aligns with our design principle of multiscale hierarchical feature alignment. Relying on a single scale fails to yield optimal localization accuracy.

3.3. Loss Factor α and β

The two hyperparameters, α and β, are to balance the localization loss and patch-matching loss in 3DFA. To identify an effective configuration, we conduct the experiments where β is fixed to be 1 and α has different values. As shown in Table 2, the best choice of α lies between 0.75 to 1.25 and we use $\alpha = 1$ in our finalized model. These results suggest that maintaining a balance between the patch-matching loss and the localization loss leads to the best overall performance. This outcome is expected: while patch matching is designed to inject spatial correspondence into the 2D and 3D feature representations, the primary objective of the framework is still accurate localization. When the patch-matching term is weighted too heavily, it can overwhelm and ultimately degrade the original point cloud features, resulting in diminished localization accuracy.

References

[1] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 1

[2] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32, 2019. 1

[3] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[4] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 1

[5] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024. 1

[6] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[7] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 1

[8] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024. 1

[9] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022. 1

[10] Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. Pnerfloc: Visual localization with point-based neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7450–7459, 2024. 1