Supplementary Materials for

CROSSFIRE: Camera Relocalization On Self-Supervised Features from an Implicit Representation

This document presents further analysis on our method. As we can not release the code, we present extensive implementation details in sec 1, in order to facilitate the reproducibility of our experiments. We also provide a more detailed ablation study on the proposed loss function in sec 2. Finally, we invite readers to view the supplementary video where visualizations of the rendered descriptors and localization results are shown and commented in sec 3.

1 Implementation and Reproducibility details

Features extractor CNN architecture We use 8 2D-convolution layers with respectively 64, 64, 64, 64, 128, 128, 128, 32 output channels and a kernel sizes of 3x3 (except the last layer, for which the kernel is 1x1). Each intermediate layer is followed by a ReLU activation and max poolings are applied after layers 2 and 4.

Neural renderer The encoding part of the neural renderer is exactly similar to Instant-NGP [4] and is implemented using tiny-cuda-nn [3] with default parameters. The appearance latent codes (used only for outdoor scenes in our experiments) have a size of 48, similar to NeRF-W [1] and the descriptor head has the same latent dimension than the RGB head, i.e. 64. Similar to NeRF [2], we first sample points linearly along the ray (coarse step) and then samples new points based on the obtained density values (fine step). Both steps are computed by the same model. We use 128 coarse and 128 fine points during training, and 256 + 256 during localization.

Scene normalization We use COLMAP [5] SfM models as input of our training pipeline. We define an axis-aligned bounding box (AABB) containing the entire scene (all camera poses + sparse point cloud). Then, we normalize all camera poses positions by dividing by their translation vectors by the largest coordinate of the scene bounding box. Rays defined for the neural renderer start from the camera center and end when they intersect the scene AABB. This way, every 3D point evaluated for rendering belongs to the unit cube.

2 Ablation study on the descriptor loss

In the main paper, we presented a qualitative comparison between the proposed descriptor loss and a classical triplet loss, which we observed to fail to learn meaningful features. In addition to that, we propose an additional experiment where we investigate the impact of the parameter λ on the proposed loss.

We train 5 models on the "Chess" scene from the 7scenes dataset [6] with different values for λ, while keeping other parameters fixed. The results are displayed in Fig 1 .

We observe that when using extreme values such as 0.1 or 10 for this parameter, the resulting model is not able to produce reliable matches and the localization process fails. When λ is small, the model learns similar descriptors for the entire scene and then is not able to effectively discriminate between areas. On the other way, when λ is too large, we ask the model to produce very different descriptors for very close points. While it could be seen as a good local property, we observe that this constraint can not be fully satisfied for a whole scene, given the compact 32-dimensional descriptors we use. A large λ parameter creates ambiguities because the same descriptor is attributed to points far from each other in the scene, resulting in wrong matches.

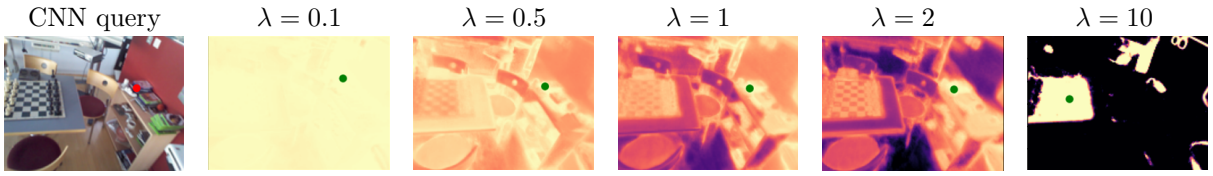

Figure 1: Similarities scores between a CNN query pixel and rendered descriptors, depending on λ. The left image shows a query pixel from a test sample while other images show a colormap of the similarity between rendered descriptors from the same viewpoint and the query pixel. Yellow color indicates a high similarity.

All experiments reported in the main paper used $\lambda = 1$, and we observe that reasonable values ($\lambda = 0.5$ and $\lambda = 2$) for this parameter results in a similar localization accuracy, suggesting that a per-scene tuning of λ is not necessary for deploying in a new environment.

Concerning localization accuracy on the test set, models trained with $\lambda = 0.1$ and $\lambda = 10$ completely fail to converge to an accurate pose, while both 3 models trained with $\lambda = 0.5, 1, 2$ have the same median error: 1cm and 0.4°.

3 Supplementary video

The first part of the supplementary video shows localization results on the test set of the chess scene. The first row shows RGB images, outputs of the neural renderer at corresponding pose, except the right image which is the query image captured by the camera. If the localization process is correct, images rendered at estimated pose should be similar to the query image. The second row shows PCA visualizations of the descriptors from the CNN and the neural renderer. The last row shows depth maps from the neural renderer. We observe that the estimated poses are very accurate on this scene thanks to similar descriptors between both models and accurate depth estimation.

The second part of the video shows similarity scores between query pixels from a test image of the "Pumpkin" scene and the rendered descriptors from the prior camera pose. We observe that all pixels belonging to the same objects have a higher descriptor similarity than the rest of the scene, which enable to establish accurate correspondences during the localization process.

References

[1] Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

[2] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)

[3] Müller, T.: tiny-cuda-nn (4 2021), `https://github.com/NVlabs/tiny-cuda-nn`

[4] Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, `https://doi.org/10.1145/3528223.3530127`

[5] Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[6] Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2930–2937 (2013)