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

Arthur Moreau[1,2], Nathan Piasco[1], Moussab Bennehar[1], Dzmitry Tsishkou[1]
Bogdan Stanciulescu[2], Arnaud de La Fortelle[2]
[1]Noah's Ark IoV team, Huawei France, [2]Mines Paris, PSL University, Centre for robotics
arthur.moreau2@huawei.com

Abstract

Beyond novel view synthesis, Neural Radiance Fields (NeRF) are useful for applications that interact with the real world. In this paper, we use them as an implicit map of a given scene and propose a camera relocalization algorithm tailored for this representation. The proposed method enables to compute in real-time the precise position of a device using a single RGB camera, during its navigation. In contrast with previous work, we do not rely on pose regression or photometric alignment but rather use dense local features obtained through volumetric rendering which are specialized on the scene with a self-supervised objective. As a result, our algorithm is more accurate than competitors, able to operate in dynamic outdoor environments with changing lightning conditions and can be readily integrated in any volumetric neural renderer.

1. Introduction

Visual localization, i.e. the problem of camera pose estimation in a known environment [36], enables to build camera-based positioning systems for various applications such as autonomous driving [28], robotics [2] or augmented reality [24]. Map-based navigation systems for such applications operate with a reference map of the environment, built from previously collected data. These maps are commonly defined with explicit 3D scenes representations (point cloud, voxels, meshes, etc.), which only store discrete information while the underlying environment they represent is continuous.

Recently, Neural Radiance Fields (NeRF) [26] and related volumetric-based approaches [31, 53] have emerged as a new way to implicitly represent a scene. 3D coordinates are mapped to volume density and radiance in a neural network. NeRF is trained with a sparse set of posed images of a scene and learns its 3D geometry via differentiable rendering. The resulting model is continuous, i.e. the radiance of
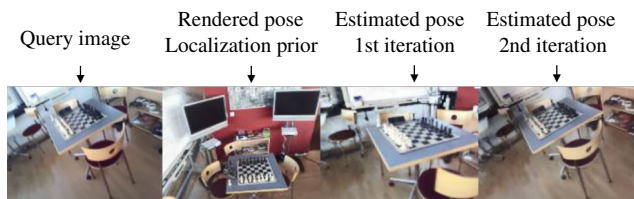

Figure 1: **Visual localization in a neural renderer.** Starting from a coarse localization prior, our algorithm estimates the pose of a query image by comparing image features to descriptors rendered from a neural scene representation.

all 3D points in the scene can be computed, which enables the rendering of photorealistic views from any viewpoint.

Beyond their rendering ability, implicit scene representations are actively investigated to be used as the map representation for navigation systems [1, 35, 20, 17]. This work focuses on one aspect of the navigation pipeline, understudied in the specific case of implicit scene representation, the image localization problem. Our motivation is to provide a camera relocalization algorithm (i.e. 6-DoF pose estimation) from one RGB image based only on a learned volumetric-based implicit map. We aim to design a method for robotics applications: it must be fast to compute, robust to outdoor conditions and could be deployed in dynamic environments. Existing localization methods that use implicit maps either have limited accuracy by lack of geometric reasoning [29, 7], or do not meet the aforementioned requirements because photometric alignment [55, 19] can be slow and assumes constant lightning conditions.

Contribution. In this paper, we introduce local descriptors in NeRF's implicit formulation and we use the resulting model, named CROSSFIRE, as the scene representation of a 2D-3D features matching method. We train simultaneously a CNN feature extractor and a neural renderer to provide consistent scene-specific descriptors in a self-supervised way. During training, we leverage the 3D in-

formation learned by the radiance field in a metric learning optimization objective which does not require supervised pixel correspondences on image pairs nor a pre-computed 3D model. The proposed descriptors represent not only the local 2D image content but also the 3D position of the observed point, which enables to solve ambiguities in areas with repetitive patterns. Our method can use any differentiable neural renderer and, hence, can directly benefit from recent NeRF improvements. For instance, we make the model computationally tractable thanks to the multi-resolution hash encoding from Instant-NGP [31] and adapted to dynamic outdoor scenes thanks to appearance embeddings from Nerf-W [25].

Finally, we show that these features can be used to solve the visual relocalization task with an iterative algorithm composed of a dense features matching step followed by standard Perspective-n-Points (PnP) camera pose computation. We take inspiration from structure-based visual localization pipelines [41, 39] but replace the commonly used sparse 3D model obtained from Structure-from-Motion by our neural field from which dense features are extracted. For a given camera pose candidate, we render dense descriptors and depth maps. Descriptors are used to establish 2D-2D matches which are upgraded to 2D-3D matches by the rendered depth. We can iteratively refine the estimated pose by repeating the aforementioned procedure, as presented in Figure 1.

2. Related work

Localization with Neural Scenes Representations. Many algorithms have recently been developed to compute the camera pose of an image w.r.t. a NeRF model.

One line of work has developed visual SLAM systems, where the implicit map is learned during the navigation. iMAP [45] and NICE-SLAM [58] leverages the depth information of RGB-D cameras to de-couple pose and scene geometry estimation. Then, NeRF-SLAM [37] extends these approaches to RGB images by using dense monocular SLAM as supervision for the NeRF map. In contrast with these methods, we target a relocalization approach, where the environment has already been visited. In this scenario, the map is pre-computed offline or derived from a SLAM approach. Our solution could be used as a relocalization module that can be plugged into implicit SLAM pipelines for continuous navigation and place re-visit.

A first relocalization solution is to align iteratively a query and a rendered image by optimizing the camera pose based on the photometric error. This has been first proposed by iNeRF [55] which demonstrates accurate pose estimation on usual NeRF datasets, i.e. controlled environments such as synthetic or static indoor scenes. However, the localization process is slow because each iteration requires rendering and backpropagation through the entire NeRF

model, and the convergence basin is small. This idea has then been improved by using more efficient rendering models and parallel optimization based on Monte-Carlo sampling [19]. Loc-NeRF [23] integrates this idea in a particle filter formulation.

Another direction uses Absolute Pose Regression [16, 28] that directly connects images and camera poses in a deep network. While these methods usually present a low accuracy [42], they can be improved by leveraging a NeRF during the training step. Direct-PoseNet [8] renders the image at the estimated pose and uses the differentiability of the renderer to define an additional loss function based on the photometric error. Then, DF-Net [7] iterates on this idea and defines a loss based on features matching. Finally, LENS [29] pre-computes a large set of synthetic views uniformly distributed across the scene and uses it as additional training data.

Related to our work, Features Query Network [14] stores local descriptors in an implicit scene representation and uses it to perform local features matching in a structure-based formulation [41, 39, 34]. While we use a related localization process, our method is novel on two crucial aspects. First, FQN is limited to a pre-computed sparse 3D point cloud, while our proposal provides dense features from a radiance field. Then, instead of memorizing in a supervised way how descriptors vary w.r.t. viewpoint in an off-the-shelf features extractor, we take the opposite direction and learn scene-specialized descriptors without supervision through a metric learning objective and decide to model these features as not dependent on the viewing direction, in order to facilitate the matching process. To the best of our knowledge, learning visual localization descriptors in a neural radiance field without supervision has not been proposed before.

Learning-based description of local features. Local descriptors provide useful descriptions of regions of interest that enable to establish accurate correspondences between pairs of images describing the same scene. While hand-crafted descriptors such as SIFT [21, 22] and SURF [5] have known great success, the focus has shifted in recent years to learn features extraction from large amounts of visual data. Many learning-based formulations [9, 15, 44, 56, 47, 11] rely on siamese convolutional networks trained with pairs or triplets of images/patches supervised with correspondences. NeRF-Supervision [54] takes advantage of the geometric consistency of depth-supervised object-centric NeRFs to obtain correspondences between different views of the object in order to learn view-invariant dense object descriptors. Features extractors can be trained without annotated correspondences by augmenting two versions of a same image or using weak supervision. SuperPoint [10] uses homographies while Novotny et al. [33] leverage image warps. In
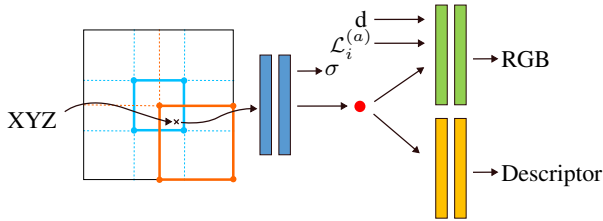
Figure 2: **Neural radiance and descriptors fields.** The input coordinate is encoded by the multi-resolution hash tables from Instant-NGP [31] enabling fast training and rendering. We use per-image appearance embeddings to handle varying illumination across training images. The descriptors heads is invariant to viewing direction and appearance vector allowing to learn robust localization features.

a recent work, CAPS [51] have shown that accurate correspondences between different views can be obtained using weak supervision through the use of relative camera poses. Our proposed method follow a different path to learn repeatable descriptors: we constraint the image feature extractor to provide the same descriptors map as the Neural Field. This approach allows us to learn dense scene-specific descriptors without annotated correspondences since the neural renderer provides similar features for rays which intersect the same point.

3. Method

The proposed algorithm estimates the 6-DoF camera pose of a query image in an already visited environment. We first train our modules in an offline step, using a set of reference images with corresponding poses, captured beforehand in the area of interest. A 3D model of the scene is not a pre-requisite because we learn the scene geometry during the training process.

3.1. Neural rendering of descriptors

Background. NeRF [26] is capable of rendering a view from any camera pose in a given scene while being trained only with a sparse set of observations. Given a camera pose with known intrinsics, 2D pixels are back-projected in the 3D scene through ray marching. The density σ and RGB color c of each point $p = (x, y, z)$ along the ray are evaluated by a MLP R_θ: $c, \sigma = R_\theta(p, d)$ where d is the viewing direction. The final pixel color of a pixel is computed with differentiable volumetric rendering along the ray, which enables to train the implicit scene representation by minimizing the photometric error of rendered images.

NeRF makes the assumption that illumination in the scene remain constant over time, which does not hold for many real world scenes. NeRF-W [25] overcomes this limitation by modeling appearance with a per-image latent

codes $\mathcal{L}_i^{(a)}$ (i.e. appearance embedding) that controls the appearance of each rendered view. Another limitation the original formulation of NeRF is the computation time: rendering an image requires $H \times W \times N$ evaluations of the 8 layers MLP, where N is the number of points sampled per ray, resulting in slow training and rendering. Recently, Instant-NGP [31] proposes to use multi-resolution hash encoding to accelerate the process by storing local features in hash tables, which are then processed by much smaller MLPs compared to NeRF resulting in significant improvement of both training and inference times.

Neural radiance and descriptors fields. CROSSFIRE combines the 3 aforementioned techniques to efficiently render dynamic scenes. However, our main objective is not photorealistic rendering but, rather, features matching with new observations. While it is possible to align a query image with a NeRF model by minimizing the photometric error [55], such approach lacks robustness w.r.t. variations in illumination. Instead, we propose to add positional features, i.e. D-dimensional latent vectors which describe the visual content of a region of interest in the scene, as an additional output of the radiance field function. In contrast with the rendered color, we model these descriptors as invariant to viewing direction d and appearance vector $\mathcal{L}_i^{(a)}$ (*i.e.* we do not provide d and $\mathcal{L}_i^{(a)}$ to the MLP head responsible of generating the positional feature, see Figure 2). We verify through ablation study in section 4.5 that this descriptor property makes the matching process more robust. Similar to color, the 2D descriptor of a camera ray is aggregated by the usual volumetric rendering formula applied on descriptors of each point along the ray. The architecture of our proposed neural renderer is summarized in Figure 2 and implementations details are provided in section. The training pipeline of CROSSFIRE is explained in the next section.

3.2. Self-supervised training of features

Motivation. In the previous section, we explained how our proposed neural renderer describes the map for relocalization purposes thanks to the introduced positional descriptors. Additionally, we also need to extract features from the query image. A simple solution, proposed by FQN [14], is to use an off-the-shelf pre-trained features extractor such as SuperPoint [10] or D2-Net [11], and train the neural renderer to memorize observed descriptors depending on the viewing direction. Optimizing scene-specific descriptors, however, allows to better differentiate repetitive patterns in the scene resulting, in a more robust localization and reducing failure cases. To this end, we propose to train jointly the feature extractor with the neural renderer by defining an optimization objective which leverages the scene geometry. We obtain descriptors specialized on the target scene which describe not only the visual content but also the 3D location
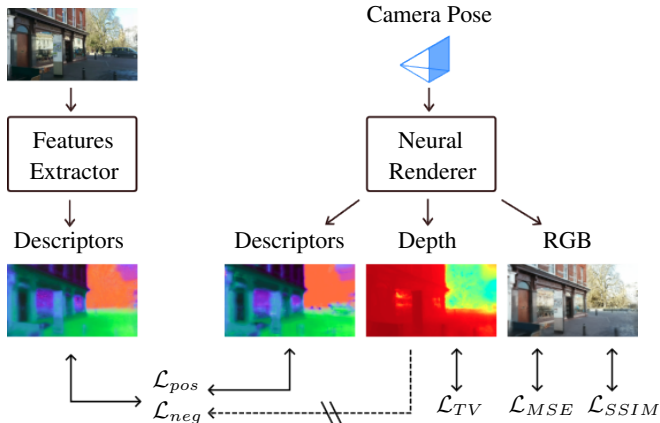
Figure 3: **Training pipeline of CROSSFIRE.** We jointly optimize the neural renderer and the features extractor to obtain robust, scene-specific localization descriptors. We use regularization losses (i.e. TV and SSIM) to increase the consistency of the neural renderer. We propose a two-terms loss that maximizes the similarity between corresponding feature maps while penalizing pixel pairs that are geometrically distant from each other.
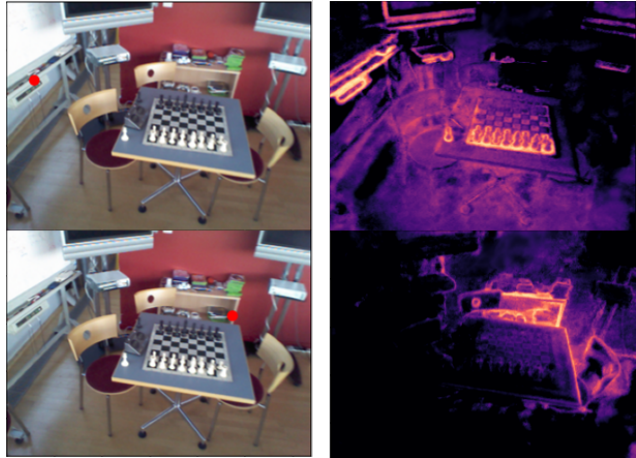

Figure 4: **Similarities of positional features.** We show the dense matching map between one descriptor from the query image (red dots in left images) and the reference descriptors from the neural renderer. Thanks to our training objective, descriptors close (in 3D) to the selected points have high similary whereas others do not match. This behaviour is enforced by our loss function.

of the observed point, with better discriminant property than generic descriptors.

The training procedure of our system is described in Figure 3. One training sample corresponds to a reference image with its corresponding camera pose. From one side, the image is processed by the features extractor to obtain the descriptors map F_I. On the other side, we sample points along rays for each pixel using camera intrinsics, compute density, color and descriptor of each 3D point, and finally perform volumetric rendering to obtain a RGB view C_R, a descriptors map F_R and a depth map D_R.

Features Extraction. Our features extractor, inspired by SuperPoint [10], is a simple fully convolutional neural network with 8 layers, ReLU activations and max poolings. The input is a RGB image I of size $H \times W$ and produces a dense descriptors map $F_I \in \mathbb{R}^{H/4 \times W/4 \times d}$.

Learning the Radiance Field. Similar to NeRF [26], we use the mean squared error loss $\mathcal{L}_{MSE}$ between C_R and the real image to learn the radiance field. As we render entire, although downscaled, images in a single training step, we can leverage the local 2D image structure and minimise the structural dissimilarity (DSSIM) loss $\mathcal{L}_{SSIM}$ [52], which we observe to produce sharper images and more accurate scene geometry. Depth maps are used by the localization process to compute the camera pose, and then better depth results in more accurate poses. NeRF models trained with limited training views can yield incorrect depths, due to the

shape-radiance ambiguity [57]. We add a regularization loss $\mathcal{L}_{TV}$ which minimizes depth total variation of randomly sampled 5x5 image patches to encourage smoothness and limit artefacts on the rendered depth maps [32]. We verify in section 4.5 that using these 3 loss functions is beneficial for the localization accuracy.

Learning the Descriptors Field. Our main goal is to match the descriptors map from the CNN features extractor and the corresponding one from the neural renderer. The self-supervised optimization objective encourages both models to produce identical features for a given pixel while preventing high matching scores between points far from each other in the 3D scene. We define a loss function with two terms $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$, applied on a pair of descriptors maps, each containing n pixels. We use the cosine similarity, noted $\otimes$, to measure similarity between descriptors.

The first loss term $\mathcal{L}_{pos}$ maximizes the similarity between descriptors maps F_I and F_R from both models:

$$\mathcal{L}_{pos} = \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - F_I[i] \otimes F_R[i]) \qquad (1)$$

The second loss term $\mathcal{L}_{neg}$ samples random pairs of pixels and ensures that pixel pairs with large 3D distances have dissimilar descriptors:

$$\mathcal{L}_{neg} = \frac{1}{mn} \sum_{k,i=1}^{m,n} \max(0, F_I[p_k(i)] \otimes F_R[i] - t_\lambda(p_k(i), i))$$

$$(2)$$

where $t_\lambda(i,j) = max(0, 1 - \lambda\|xyz(i) - xyz(j)\|)$. $xyz(i)$ is the 3D coordinate of the point represented by the i_{th} pixel in the descriptors map. We compute it from the camera parameters of the rendered view and predicted depth. It should be noted that we do not backpropagate the gradient of this loss to the depth map because the gradient of this loss does not provide meaningful signal to learn the scene geometry. λ is an hyperparameter which controls the maximum similarity between descriptors at a given 3D distance. $(p_k)_m$ are random permutations of pixel indices from 1 to n.

The proposed self-supervised objective is close to a classical triplet loss [3], but we show in Figure 9 that scaling the loss by the 3D coordinates in the formulation is crucial to learn smooth and selective descriptors. A visualization of the similarity between descriptors enforced by the proposed loss is shown in Figure 4.

Finally, we optimize the following loss function at each training step:

$$\mathcal{L} = \mathcal{L}_{MSE} + \lambda_1 \mathcal{L}_{SSIM} + \lambda_2 \mathcal{L}_{TV} + \mathcal{L}_{pos} + \mathcal{L}_{neg} \quad (3)$$

where $\lambda_1 = 0.1$ and $\lambda_2 = 1e^{-3}$ are hyper-parameters introduced to balance SSIM and TV losses, respectively.

3.3. Visual Localization by iterative dense features matching

This section describes the localization pipeline used to estimate the camera pose of a given query image using our learned renderer and features. An overview of this procedure is shown in Figure 5. The proposed solution combines simple and commonly used techniques and we do not claim algorithmic novelty on this part. The goal is, rather, to demonstrate that the quality and robustness of our learned features enables to reach precise localization while using basic features matching and pose estimation strategies.

1. Localization prior. Similar to related features matching methods [41, 39, 14], we assume to have access to a localization prior, i.e. a camera pose relatively close to the query pose. A view observed from the prior should have an overlapping visual content with the query image to make the matching process feasible. Such priors can be obtained by matching a global image descriptor against an image retrieval database [3, 39] or an implicit map [27].
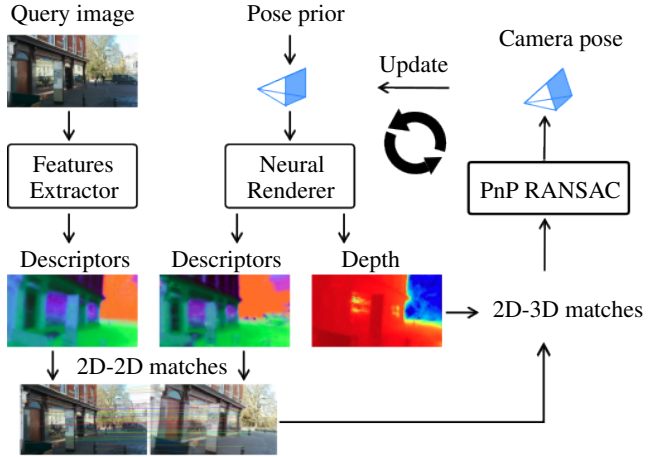

Figure 5: **CROSSFIRE localization procedure.** Descriptors are extracted from the query image and matched against descriptors rendered from the localization prior. Depth information provides 2D-3D matches that enable to compute the pose with PnP + RANSAC. This process can be repeated iteratively, by rendering descriptors from the predicted pose.

2. Features extraction. First, we extract dense descriptors from the query image through the CNN. On the other side, descriptors and depth corresponding to the localization prior are computed by the neural renderer.

3. Dense Features Matching. Query and reference descriptors are matched with cosine similarity. We consider that 2 descriptors are a match if the similarity is higher than a threshold θ and if it represent the best candidate in the other map in both direction (mutual matching). We then compute the predicted 3D coordinate of rendered pixels which have been matched (thanks to camera parameters and depth) and obtain a set of 2D-3D matches.

4. Camera Pose Estimation. To compute the camera pose from the 2D-3D matches, we use the Perspective-N-Points algorithm combined with RANSAC [13], in order to get a robust estimate by discarding outliers matches.

5. Iterative Pose Refinement. While classical 3D models only have access to a finite set of reference descriptors, our neural renderer can compute them from any camera pose. Similar to FQN [14] and ImPosing [27], we can then consider the camera pose estimate as a new localization prior and iterate the previously mentioned steps multiple times to refine the camera pose.
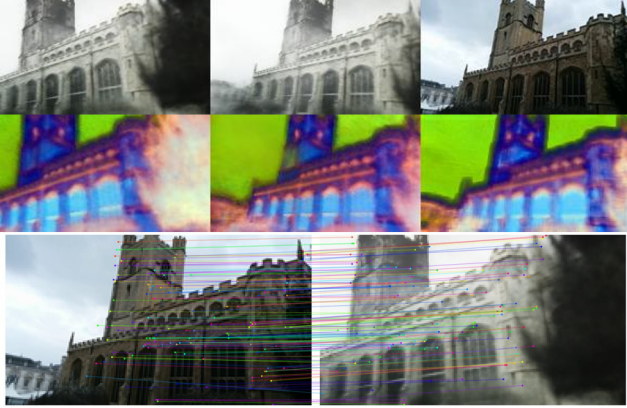
Figure 6: **Visualization of rendered views, descriptors and matches in StMarysChurch.** We show on the top row the query image (right), the RGB rendered view from the localization prior (left) and from the 1st estimated pose (middle). The second row represents a PCA visualization of the corresponding descriptors map from the neural renderer (left and middle) and the features extractor (right). The last row displays the inlier matches obtained by our pipeline.

4. Experiments

We first present a comparison of CROSSFIRE with related methods relocalization that use implicit map representations in section 4.1. We also evaluate the impact of the localization prior in section 4.2 and additional ablation studies in 4.5.

Implementation. Our system is implemented in PyTorch. The hash tables and MLPs of the neural renderer use tiny-cuda-nn [30]. We use the default PnP pose solver from PoseLib [18]. In all the proposed experiments, we use descriptors of size 32. We train the models for 100k iterations. The initial learning rate is set to $1e^{-3}$ and reduced to $1e^{-4}$ after 2000 iterations. For ensuring reproducibility, the detailed architecture of our neural networks are provided in supplementary materials.

Datasets. We evaluate our method on 2 standard localization benchmarks. 7scenes [43] consists in indoor static scenes captured using a hand-held camera. Cambridge Landmarks [16] contains outdoor scenes representing buildings observed from different viewpoints and lighting conditions, with dynamic occluders such as pedestrians and cyclists in both train and test sets.

Efficiency. The storage requirement of our modules is 50MB (48MB for the hash tables and 2MB for the neural networks). In contrast with explicit maps, this number does not grow with the amount of reference data. All trainings

and inferences have been performed on a RTX3090 GPU. Trainings take approximately 5 hours for indoor scenes and 15 hours for larger outdoor scenes. Inference times are: 9ms for features extraction, 5ms for rendering, 5ms for dense matching and ≈ 60ms for PnP+RANSAC (because we have a lot of matches), resulting in ≈ 200ms for the total time with 3 iterations reported in the experiments. Speedup can be achieved easily by less refinements, at the cost of minor accuracy drop.

4.1. Comparison to related methods

We evaluate our method on both datasets using a maximum of 3 iterations of the localization process. We use as localization prior the top 1 reference pose retrieved by DenseVLAD [48]. In order to render reference frames efficiently, the matching step is done at a small resolution: 194x108 for Cambridge Landmarks and 161x120 for 7scenes.

We compare our algorithm to the learning-based visual relocalization methods that use implicit map representations in their pipeline.

- Direct-PoseNet [8] train an Absolute Pose Regressor with an additional photometric loss by rendering the estimated pose through NeRF.

- DFNet [7] goes in the same direction but defines a features matching loss with the rendered view.

- LENS [29] trains an absolute pose regressor with NeRF rendered views uniformly distributed across the scene.

- FQN [14] regresses descriptors in an implicit representation of a sparse 3D model. This method is the closest to our work because it uses the same iterative localization process and store descriptors in a neural scene representation. The main differences are that descriptors are not trained specifically from the scene but memorized from a pretrained features extractors, and that the representation is sparse whereas ours is dense. Results are reported for D2-Net [11] and MobileNetv2 [38] descriptors.

iNerf [55] and related methods are not present in our evaluation, first because results on usual localization benchmarks are not reported in the corresponding papers, but also because it does not meet the robotics requirements described before, i.e. fast inference for iNeRF and compatibility with outdoor dynamic environments.

The results of the comparisons for both datasets are shown in Table 1. CROSSFIRE obtains the lowest error for both indoor localization and outdoor scenes. Results on the highly ambiguous Stairs scene are higher than in other
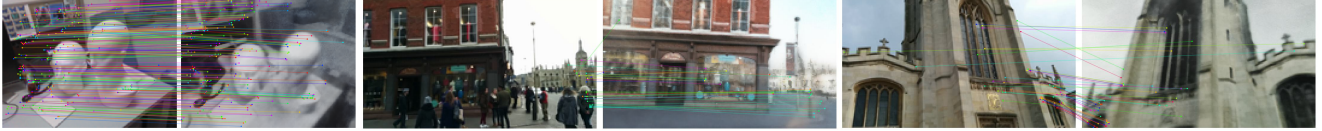
Figure 7: **Success and failure cases:** we show inliers matches between the query image and the NeRF rendered image at prior pose. Using dense features field for localization enables to establish accurate correspondences in texture-less areas (left). Failure cases are observed in the presence of dynamic objects (middle), for which the PnP converges on a wrong pool of matches, and ambiguous cases (right) where the CNN mixes up the symmetrical parts of the church due to lack of long-range reasoning.

Dataset / Methods	Absolute Pose Regression + NeRF			Implicit local features		
Cambridge	DirectPN [8]	DFNet [7]	LENS [29]	FQN-D2N [14]	FQN-MN [14]	**CROSSFIRE (Ours)**
Kings College	-	0.73m / 2.4°	0.33m / 0.5°	0.32m / 0.5°	**0.28m / 0.4°**	0.47m / 0.7°
Old Hospital	-	2.00m / 3.0°	0.44m / 0.9°	0.64m / 0.9°	0.54m / 0.8°	**0.43m / 0.7°**
Shop Facade	-	0.67m / 2.2°	0.27m / 1.6°	0.14m / 0.6°	**0.13m / 0.6°**	0.20m / 1.2°
StMarys Church	-	1.37m / 4.0°	0.53m / 1.6°	0.93m / 3.5°	0.58m / 2.0°	**0.39m / 1.4°**
Average	-	1.19m / 2.9°	0.39m / 1.2°	0.51m / 1.4°	0.38m / 1.0°	**0.37m / 1.0°**
7scenes						
Chess	0.10m / 3.5°	0.05m / 1.9°	0.03m / 1.3°	0.06m / 1.9°	0.04m / 1.3°	**0.01m / 0.4°**
Fire	0.27m / 11.7°	0.17m / 6.5°	0.10m / 3.7°	0.14m / 4.1°	0.10m / 3.0°	**0.05m / 1.9°**
Heads	0.17m / 13.1°	0.06m / 3.6°	0.07m / 5.8°	0.05m / 3.5°	0.04m / 2.4°	**0.03m / 2.3°**
Office	0.16m / 6.0°	0.08m / 2.5°	0.07m / 1.9°	0.14m / 4.1°	0.10m / 3.0°	**0.05m / 1.6°**
Pumpkin	0.19m / 3.9°	0.10m / 2.8°	0.08m / 2.2°	0.10m / 2.6°	0.09m / 2.4°	**0.03m / 0.8°**
Kitchen	0.22m / 5.1°	0.22m / 5.5°	0.09m / 2.2°	0.18m / 4.8°	0.16m / 4.4°	**0.02m / 0.8°**
Stairs	0.32m / 10.6°	0.16m / 3.3°	0.14m / 3.6°	1.41m / 53.0°	1.40m / 34.7°	**0.12m / 1.9°**
Average	0.20m / 7.3°	0.12m / 3.7°	0.08m / 3.0°	0.30m / 10.6°	0.28m / 7.3°	**0.04m / 1.1°**

Table 1: **6-DoF median localization errors of visual localization methods based on implicit representations.** Direct-PoseNet did not report results for Cambridge Landmarks.

scenes but still better than other methods for which the localization process sometimes totally fail.

Furthermore, we consistently perform better than NeRF-assisted APR methods and, more importantly, than pre-trained implicit descriptors. Because the camera pose estimation process used in FQN is similar than in ours, these results indicate that our scene-specific features are beneficial compared to off-the-self features extractors.

We hypothesize that the absolute localization accuracy in outdoor scenes is lower for 2 main reasons. First, we lack a way to handle dynamic content such as pedestrians during the test step, which we observe to degrade the quality of our matches. Second, the quality of depth maps in these scenes is less accurate than in indoor scenarios, especially for background, due to observable image content very far from the camera. As we use depth to compute the 3D coordinates of matches, this introduces noise in the localization process.

4.2. How good the pose priors need to be?

To measure how bad initialization impacts localization results, we conducted an experiment on the Chess scene where we replace the prior from image retrieval by using

the same prior for all test images (shown in Figure 1). Results are shown in Table 2. We observe that, thanks to our iterative refinement, imprecise priors do not affect the final localization accuracy but rather require more iterations to reach the correct camera pose.

cm / °	Prior	Iter 1	Iter 2	Iter 3
Retrieval	0.22 / 12.1	0.02 / 0.7	0.01 / 0.5	0.01 / 0.4
Constant	1.82 / 32.2	0.12 / 2.8	0.02 / 0.6	0.01 / 0.5

Table 2: **Impact of prior accuracy:** Median error w.r.t. prior strategy and iterations.

4.3. Evaluation of the features extractor

Beyond the localization accuracy of the entire method, we conducted an experiment to compare the matching accuracy of our scene-specialized features extractor to those of SuperPoint [10], which is a popular pre-trained learning-based method. Because we need to train a neural field on each scene, we can't use the HPatches benchmark [4] for such purpose. On the Chess scene, we compute reference matches between test images thanks to NeRF geometry.
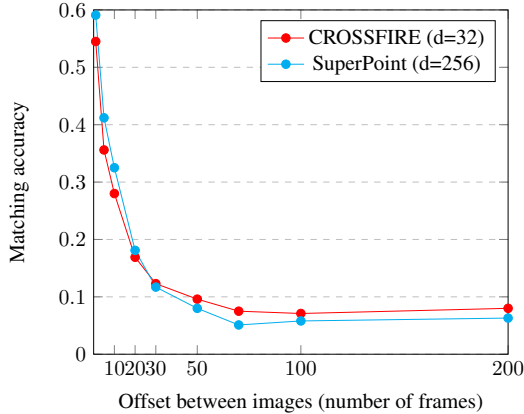
Figure 8: **Comparison between features extractors.** We plot the matching accuracy on the Chess scene depending on the time offset between images.

Then, we compare it with predicted matches to compute the matching accuracy (or precision). We compute this for image pairs with varying time offsets and report results in Figure 8. We observe that SuperPoint descriptors enable better matching when the viewpoint is close, but CROSS-FIRE is more accurate with large viewpoint discrepancy. It should be noted that CROSSFIRE descriptors are 8 times more compact (32 vs 256).

4.4. Qualitative evaluation

We provide visualization of success and failure cases in Figure 7. In the first failure case in the Shop Facade scene (middle), we observe that the set of inliers matches is entirely incorrect, but, probably out of bad luck, consistently lead to a (wrong) camera pose. The RANSAC loop selected this pool of correspondences that lies in pedestrians instead of other matches on the shop. This problem could be addressed by confidence estimation, that we leave as future work. For the second case, the only way to distinguish the left side from the right side of the church is to reason on the entire image, since symmetrical parts are locally similar. Because the confusing areas are far from each other in the image and our CNN uses small convolutional filters, such long-range reasoning is prevented and features from the right side in the query are wrongly matched with the left side. This could be improved with attention mechanisms in the features extractor architecture.

More visualizations are provided in Figure 6 and in the supplementary video.

4.5. Ablation studies

Descriptor loss. The self-supervised loss used to train descriptors is similar to the triplet loss commonly used for metric learning, except an additional term for negative pairs

Figure 9: **Qualitative comparison of descriptors between the proposed loss and a classical triplet loss.** We visualize the PCA of descriptors from our loss (middle) and a triplet (right) for a given query image (left).

which depends on the 3D distance between points. We propose a qualitative comparison between the triplet loss and our proposal in Figure 9. We observe that the representation learned by our system is smooth and more expressive than the triplet loss which only separate the scene into few clusters. More details including a quantitative comparison is provided in supplementary materials.

Conditioning descriptors with viewing direction. We modeled the descriptors learned by the neural renderer as independent of the direction from which the point is observed. We verify that this choice is relevant by comparing it to the view-dependent case. Modeling the descriptors as dependent on the image appearance is not feasible because this parameter is unknown during the localization step. The comparison is shown in Figure 10.

Reconstruction losses. We evaluated the benefits of the $\mathcal{L}_{SSIM}$ and $\mathcal{L}_{TV}$ terms of the loss function on the localization accuracy on Figure 11. On the Heads scene, the error is 3cm/2.3° with the proposed loss, 4cm/2.1° without $\mathcal{L}_{SSIM}$ and 6cm/4.0° without $\mathcal{L}_{TV}$. These terms actually improve the localization accuracy because they help to recover the correct scene geometry.
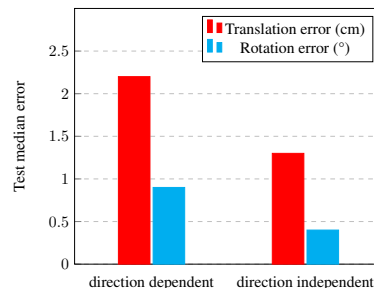

Figure 10: **Localization accuracy depending on descriptor head inputs.** We compare the final accuracy on the "Chess" scene with and without the viewing direction as descriptor input in the neural renderer.
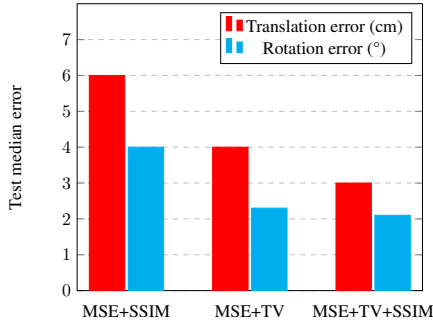
Figure 11: **Impact of additional reconstruction losses on localization accuracy.** Translation and orientation error for several combinations of loss terms.

5. Limitations and Future Work

Scalability. Similar to other Neural Scene Representations, our Neural Field struggles to represent large scale maps, such as the one used in autonomous driving, with a single radiance field instance. The current best solution, proposed by Block-NeRF [46], is to split the environment into several smaller neural fields and enforce consistency at their boundaries. This solution is successful at a city-scale and could be implemented in our method for large scale localizatoin.

Localization pipeline. The proposed localization algorithm could be improved in many ways. Dense features matching could be performed by learning-based approaches [49, 6, 12] instead of simple heuristics. Resulting 2D-3D matchs could be improved by co-visibility filtering [41, 34]. Finally, the estimated camera pose could be optimized by direct features alignment, similar to GN-Net [50] and PixLoc [40]. The contribution of this paper lies in the learning of descriptors in a neural renderer, and this proposal can be used as a backbone for different and more advanced localization solutions.

6. Conclusion

We propose CROSSFIRE; a new way to learn and represent visual localization maps based on neural radiance fields. The proposed formulation has the advantage of densely representing local features of a scene in a compact way, and to be more robust to lightning changes than photometric alignment. We demonstrate that the non-supervised learned local features, which are specialized on the target area, perform better than related supervised techniques that use pre-trained features. The proposed implicit representation can serve as a backbone to more advanced features matching pipelines and should be compatible with future improvements in the neural rendering field that could en-

able to scale these models to larger scenes and yield better localization accuracy by improving further the quality of the learned scene geometry. We believe that replacing classical data structures by implicit scenes representations is an exciting research direction for the whole area of 3D computer vision as it enables to store dense information in a compact representation.

Acknowledgments

References

[1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[2] Yusra Alkendi, Lakmal Seneviratne, and Yahya Zweiri. State of the art in vision-based localization techniques for autonomous navigation systems. *IEEE Access*, 9:76847–76874, 2021.

[3] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[6] Gabriele Berton, Carlo Masone, Valerio Paolicelli, and Barbara Caputo. Viewpoint invariant dense matching for visual geolocalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 12169–12178, October 2021.

[7] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[8] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *2021 International Conference on 3D Vision (3DV)*, pages 1175–1185. IEEE, 2021.

[9] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS*. Curran Associates, Inc., 2016.

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Deep Learning for Visual SLAM Workshop*, 2018.

[11] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Polle-feys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Lo-cal Features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17765–17775, June 2023.

[13] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analy-sis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[14] Hugo Germain, Daniel DeTone, Geoffrey Pascoe, Tan-ner Schmidt, David Novotny, Richard Newcombe, Chris Sweeney, Richard Szeliski, and Vasileios Balntas. Feature query networks: Neural surface description for camera pose refinement. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5067–5077, 2022.

[15] Michael Jahrer, Michael Grabner, and Horst Bischof. Learned local descriptors for recognition and matching. In *Computer Vision Winter Workshop*, volume 2, pages 103–118, 2008.

[16] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolu-tional network for real-time 6-dof camera relocalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

[17] Obin Kwon, Jeongho Park, and Songhwai Oh. Renderable neural radiance map for visual navigation. *arXiv preprint arXiv:2303.00304*, 2023.

[18] Viktor Larsson. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020.

[19] Yunzhi Lin, Thomas Müller, Jonathan Tremblay, Bowen Wen, Stephen Tyree, Alex Evans, Patricio A Vela, and Stan Birchfield. Parallel inversion of neural radiance fields for robust pose estimation. *arXiv preprint arXiv:2210.10108*, 2022.

[20] Yen-Chen Lin, Pete Florence, Andy Zeng, Jonathan T Bar-ron, Yilun Du, Wei-Chiu Ma, Anthony Simeonov, Al-berto Rodriguez Garcia, and Phillip Isola. Mira: Mental im-agery for robotic affordances. In *6th Annual Conference on Robot Learning*, 2022.

[21] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157. Ieee, 1999.

[22] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vi-sion*, 60(2):91–110, 2004.

[23] Dominic Maggio, Marcus Abate, Jingnan Shi, Courtney Mario, and Luca Carlone. Loc-nerf: Monte carlo local-ization using neural radiance fields. In *2023 IEEE Inter-national Conference on Robotics and Automation (ICRA)*, pages 4018–4025. IEEE, 2023.

[24] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: A hands-on survey.

[25] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duck-worth. NeRF in the Wild: Neural Radiance Fields for Uncon-strained Photo Collections. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view syn-thesis. In *Proceedings of the European Conference on Com-puter Vision (ECCV)*, 2020.

[27] Arthur Moreau, Thomas Gilles, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Imposing: Implicit pose encoding for efficient visual local-ization. In *Proceedings of the IEEE/CVF Winter Confer-ence on Applications of Computer Vision*, pages 2892–2902, 2023.

[28] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bog-dan Stanciulescu, and Arnaud de La Fortelle. Coordinet: uncertainty-aware pose regressor for reliable vehicle local-ization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2229–2238, 2022.

[29] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localiza-tion enhanced by nerf synthesis. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1347–1356. PMLR, 2022.

[30] Thomas Müller. tiny-cuda-nn, 4 2021.

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexan-der Keller. Instant neural graphics primitives with a multires-olution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[32] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Reg-nerf: Regularizing neural radiance fields for view synthe-sis from sparse inputs. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5480–5490, 2022.

[33] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *Proceedings of the Conference on Computer Vision and Pattern Recogni-tion (CVPR)*, pages 3637–3645, 2018.

[34] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. MeshLoc: Mesh-Based Visual Localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[35] Michael Pantic, Cesar Cadena, Roland Siegwart, and Lionel Ott. Sampling-free obstacle gradients and reactive planning in neural radiance fields. In *Workshop on" Motion Planning with Implicit Neural Representations of Geometry" at 2022 IEEE International Conference on Robotics and Automation (ICRA 2022)*, 2022.

[36] Nathan Piasco, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet. A survey on visual-based localization:

IEEE Transactions on Visualization and Computer Graph-ics, 22(12):2633–2651, 2016.

On the benefit of heterogeneous data. *Pattern Recognition*, 74:90–109, 2018.

[37] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022.

[38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[39] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[40] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning robust camera localization from pixels to pose. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[41] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[42] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-taixé. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3302–3312, 2019.

[43] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013.

[44] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 118–126, 2015.

[45] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[46] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. *arXiv*, 2022.

[47] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11016–11025, 2019.

[48] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817, 2015.

[49] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. GOCor: Bringing globally optimized correspondence volumes into your neural network. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS*, 2020.

[50] Lukas Von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. GN-Net: The gauss-newton loss for multi-weather relocalization. *IEEE Robotics and Automation Letters*, 5(2):890–897, 2020.

[51] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[52] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[53] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022.

[54] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. NeRF-Supervision: Learning dense object descriptors from neural radiance fields. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2022.

[55] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021.

[56] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4353–4361, 2015.

[57] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020.

[58] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12786–12796, June 2022.