IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Visual localization using implicit representations and particle filtering-based pose refinement

GRIGORIOS-ARIS CHEIMARIOTIS[1] and DIMITRIOS ZARPALAS[1]

[1]Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), Thessaloniki, Greece (e-mail: acheimar@iti.gr, zarpalas@iti.gr)

Corresponding author: Grigorios-Aris Cheimariotis (e-mail: acheimar@iti.gr).

This work was supported by the European Union's Horizon Europe programme under grant number 101092875 "DIDYMOS-XR" (https://www.didymos-xr.eu).

ABSTRACT This work proposes HAL-NeRF v2, a localization pipeline that couples direct pose regression with Monte Carlo–based refinement on neural scene representations. Building upon the original HAL-NeRF framework, the proposed system leverages Gaussian Splatting for fast, high-detail synthetic view generation during pose regressor training, and NeRFs for efficient pose refinement. The refinement stage is redesigned with Cauchy loss, systematic resampling, and maximum-likelihood estimation,instead of the HAL-NeRF steps, to improve convergence stability in scenes with transient elements and/or incomplete mapping. In addition, HAL-NeRF v2 evaluates multiple rendered views in parallel during refinement, allowing the particle filter to rapidly disambiguate candidate viewpoints. The experimental results in the Cambridge Landmarks dataset demonstrate an accuracy equivalent to that of HAL-NeRF v1 (0.09 m / 0.61° median translational/rotational error) while achieving a 25× reduction in run time.

INDEX TERMS Gaussian Splatting,Monocular Localisation, NeRFs ,Particle Filter, Pose Refinement, Visual Localisation,

I. INTRODUCTION

LOCALIZATION of a robot or camera is crucial for extended reality applications, robotics [1] and autonomous driving [2], as it enables for precise interaction with the environment and can serve as an alternative for detecting vehicles or robots when other means fail e.g. GPS. This task i.e. estimation of the six-degree-of-freedom (6-DoF) pose of a camera from its corresponding 2D capture without input from the more expensive depth sensors, is commonly named monocular camera re-localization or visual localization.

Accurate visual localization remains a challenge, especially in terms of position (translational error). Several deep learning methods [3] [4] [5] [6] [7] [8] [10] [11] directly regress camera poses from RGB inputs, using a convolutional backbone trained end-to-end. Although pose regression models, which fall also into the Absolute pose regression(APR) category, are computationally efficient, they typically lack geometric reasoning, leading to limited generalization across viewpoints or illumination changes.

Another option is the estimation of a camera's position and orientation within a specified coordinate system, typically referenced to a pre-computed map. In this context, estimating a camera's pose from a single image requires representations that are both geometrically consistent and photometrically rich. For many methods [12] [13] [4] [14] [15] [16], the choice of the precomputed map is a NeRF [17]. NeRFs learn continuous volumetric representations that enable photorealistic novel view synthesis and can indirectly support localization tasks by comparing rendered and observed views. Gaussian Splatting [18] has emerged as a powerful alternative to NeRF, offering orders-of-magnitude faster rendering through explicit 3D point primitives with anisotropic Gaussian kernels. Therefore, a Gaussian Splatting [18] representation is another option that can replace NeRFs in these methods.

DFNet [4] and Loc-NeRF [13] are among the methods that effectively used NeRFs for visual localization. The DFNet method, which is an APR CNN-based pose regressor, included a histogram assisted NeRF (NeRFh) that preserved the appearance of the views of each pose and used it to augment the training data set by synthesizing new pairs of poses and images. DFnet with direct-feature matching further enhanced DFnet's results, achieving median translational error of 50 cm and median rotational error of 0.67 degrees in the most challenging scene of those which are typically used as benchmarks.

Loc-NeRF used vanilla NeRF [17] to employ a particle filtering localization scheme, where candidate particles(poses) are compared against the query image, and the particle filter is

updated according to the photometric loss. However, standard NeRFs are computationally demanding and ill-suited for real-time application of particle filtering.

In HAL-NeRF [19], DFnet and Loc-NeRF were sequentially combined: a) a coarse pose estimation calculated by DFnet was input to b) the particle filtering scheme . HAL-NeRF used nerfacto [23] as its NeRF representation for both modules, i.e. pose regressor (for data augmentation) and particle filter pose refinement. The full pipeline achieved fine results, i.e. very low translational and rotational errors, but with the expense of processing time. The pose regressor inferred a pose estimation instantly , however, particle filter optimization achieved its best results after fifty iterations attributing to 26 seconds of processing time. While being faster than standard vanilla NeRF, nerfacto maintains a balance between photorealism and rendering time. In addition, HAL-NeRF's implementation rendered pixels from multiple candidate viewpoints sequentially. That is, for each candidate pose (particle), the corresponding rays were independently traced through the NeRF, resulting in a per-particle rendering loop.

Building upon these insights, the present work introduces key enhancements designed to increase efficiency, reduce processing time, and establish a more consistent experimental protocol. In this paper, we propose HAL-NeRF v2, an extension of the original two-stage localization scheme whose goal is to improve its performance, mainly in terms of processing time, achieving nearly real-time pose estimation. The processing time was significantly reduced (10 times) by passing all candidate viewpoints through the MLP at once. Furthermore, the steps of Monte Carlo optimization were reconfigured : a) systematic resampling instead of multinomial resampling to keep a more diverse set of candidate poses, b) Cauchy loss was chosen to account for transient objects that are not present in rendered images, and c) instead of getting the mean of particles as the pose estimation, the pose that presents lower loss, is proposed as each step's pose estimation. These changes reduced the number of iterations needed to achieve fine results to 10. The proposed method focuses on localizing a single query image within a known scene, which enables direct comparison with existing NeRF-based localization approaches and avoids assumptions about temporal continuity.

The key contributions of this paper are as follows:

Improvement of the HAL-NeRF pipeline by providing a faster localization scheme, mainly due to multi-view parallel processing of candidate poses.

Demonstration of the effect of high-detail neural renderings for data augmentation for view-based pose regression.

An updated (compared to HAL-NeRF) Monte Carlo–based pose refinement approach with systematic resampling instead of multinomial resampling, maximum probability estimation instead of weighted average estimation, Cauchy loss instead of L2 loss. The updates in pose refinement led to more accurate results after 10 iterations.

These improvements collectively advance the practicality of neural-rendering-based localization, particularly in scenarios where rapid hypothesis evaluation is required.

The rest of the paper is structured as follows: in II Related Work,related visual localization methods are discussed. Then, in III Methods, the proposed methodology is described. In IV Results, the application of the methodology is presented. Finally, in the V Discussion, strengths, limitations, and future directions are examined.

II. RELATED WORK

A. DIRECT CAMERA POSE REGRESSION

A prominent group of techniques for visual localization, is Absolute Pose Regression (APR). Several APR methods are deep-learning pose regression methods [3] [5] [6] [7] [8] [10] [11], which aim to estimate 6-DoF poses directly from RGB inputs. PoseNet [3] introduced a convolutional neural network trained with geometric loss functions to jointly predict translation and rotation. Later works such as [10] and LSTM-Pose [9] improved performance through spatial-temporal consistency, direct feature matching, and structural regularization. While APR methods offer the advantage of direct pose estimation with instant inference, they often face challenges in achieving high accuracy due to factors such as limited training data and complex scene variations. Direct regression networks often lack geometric reasoning and struggle to generalize beyond the training domain, particularly when illumination or viewpoint changes occur.

To overcome these, other methods, e.g. [21], [4], [12], leverage end-to-end architectures to extract pose-related information from implicit representations. A critical aspect of such localization methods is the construction of a robust map that can effectively support the localization pipeline. The quality of the map is essential to ensure accurate and reliable position estimation. Thus, Neural Radiance Fields are a popular option.

B. NERF REPRESENTATION

Neural Radiance Fields (NeRF) [17], modern implicit scene representations, learn continuous volumetric representations that map 3D coordinates and view directions to color and density, enabling photorealistic view synthesis. NeRFs enable the synthesis of novel views from the learned scene representations. This capability is useful for augmenting training datasets. Recently, rapid advancements in Neural Radiance Fields (NeRFs) have positioned them as a widely adopted component in localization pipelines. Nerfacto [23] is a NeRF implementation that combines advantages of other NeRF methods.

C. NERF-BASED LOCALISATION METHODS

Several studies have used NeRF representations for localization by minimizing photometric or feature-space errors between rendered and observed views [12], [4], [26], [24] and [25]. Beginning with an initial pose estimate, iNeRF [12] used a NeRF model to estimate 6 DoF pose for scenes and objects with complex geometry by applying gradient descent

to minimize the discrepancy between pixels rendered by a NeRF and those in the observed image. LATITUDE [24] follows a similar approach to iNeRF by incorporating an initial pose prediction; however, it diverges in the optimization stage by employing a Truncated Dynamic Low-pass Filter, which helps prevent the optimization process from becoming trapped in local optima.

DFnet distinguished itself by leveraging direct feature matching, in contrast to conventional photometric loss approaches. CROSSFIRE [26] and NerfMatch [16] presented an implicit representation for local descriptors, which facilitates iterative 2D-3D feature matching. NeFeS [15] proposed a test-time refinement pipeline that improves APR methods by leveraging implicit geometric constraints through a robust feature field, allowing better utilization of 3D information during inference. A similar approach is presented in [25], where a novel neural volumetric pose feature, termed PoseMap, is introduced. However, in that work, PoseMap is integrated directly into the APR model instead of being used as a refinement method.

LENS [14] applied novel view synthesis using Nerf in the Wild [22] renderings. The synthetic data set allowed for improved camera pose regression accuracy. DFNet [4] also applied data augmentation using their own histogram assisted NeRF (NeRFh). This strategy improved pose regression accuracy through semi-supervised learning on unlabeled data. HAL-NeRF also used novel view synthesis (NVS), when training its pose regressor but using nerfacto instead of NeRFh. In this extension of HAL-NeRF, splatfacto is used instead of nerfacto to enable faster training.

D. MONTE CARLO LOCALIZATION USING NERFS

Monte Carlo Localization (MCL) [13] is a probabilistic framework that represents the pose uncertainty through a set of weighted particles. Each particle hypothesizes a possible camera pose and is iteratively refined through weighting and resampling based on observation likelihoods. Leveraging Monte Carlo localization as a core mechanism for pose estimation with a NeRF map model, Loc-NeRF [13] claims faster localization and operates without the need for an accurate initial pose estimate. Loc-NeRF [13] used L2 photometric loss between NeRF-generated views and a query image to update a particle filter optimization scheme.

HAL-NeRF adopted this Monte Carlo technique but used it in a pose refinement setup. However, this approach remained computationally heavy due to NeRF's volumetric sampling and limited rendering speed, making real-time or large-scale deployment challenging. Therefore, in the extension of HAL-NeRF presented here, Monte Carlo refinement is modified with the most impactful changes being: a) passing rays of multiple views through nerfacto's MLP for speed-up, b)a robust Cauchy loss to handle photometric inconsistencies at the presence of transient objects, and c) using maximum likelihood estimation instead of the weighted average estimate.

E. GAUSSIAN SPLATTING AND FAST RADIANCE REPRESENTATIONS

3D Gaussian Splatting [18] replaces volumetric ray marching with explicit anisotropic Gaussian primitives that can be projected and accumulated efficiently in image space. This representation allows for high-fidelity novel view synthesis in real time and has rapidly become a standard alternative to NeRF for 3D reconstruction. Splatfacto [23] [18] extends this concept by integrating Gaussian Splatting into the Nerfstudio pipeline, combining high visual quality with optimized GPU rendering. Despite the strong geometric consistency and real-time capability of Gaussian Splatting, its use for camera pose estimation or localization remains relatively unexplored. Our work uses splatfacto for view-based data augmentation in the DFnet training pipeline, in place of nerfacto that was used in HAL-NeRF and in place of NeRFh, that was used in DFnet. Thus, leveraging the splatfacto rendering efficiency, multiple pairs of synthetic images and poses are generated.

III. METHODS

A. OVERVIEW

An overview of the pipeline is presented in Figure 1. It consists of two main components: pose regression and pose refinement. A Neural Radiance Field (NeRF) is trained to implicitly represent the scene. Using this trained NeRF, a set of synthetic images is generated along real observed images to train a pose regressor that predicts camera poses.

In the first component, an initial pose estimate is obtained from a specialized CNN regressor (DFnet). In the second module, the pose is further refined using a Monte Carlo particle filter, inspired by the Loc-NeRF approach [13]. The trained part is the NeRF representation and the pose regressor. The training and test sets were divided according to [3] and the test images are used only to infer a first pose estimation using DFnet which is then refined by the pose refinement module. The refinement process involves adjusting the weights of the particles(poses) by comparing the photometric loss between the rendered images at each particle's pose and the query image, which improves the accuracy of the initial pose prediction. This component is not trained and runs at test time. In the pose refinement part, the modified parts of "HAL-NeRF v1", were the use of Cauchy loss instead of L2 loss, the use of maximum likelihood estimation instead of weighted average estimation, the systematic resampling instead of multinomial, and the parallel processing of multiple views instead of sequential.

B. POSE REGRESSOR

In this section, we refer to the Pose Regressor of Figure 1 that gives the first pose estimate that is then further refined by Monte Carlo optimization. In HAL-NeRF and in its extension presented here, DFnet with direct feature matching [4], a relatively high-accuracy pose regressor, was adopted for coarse pose estimation.

Dfnet [4], builds on PoseNetV2 [3] (they both have VGG16 [27] as the backbone). It extracts multi-scale features from

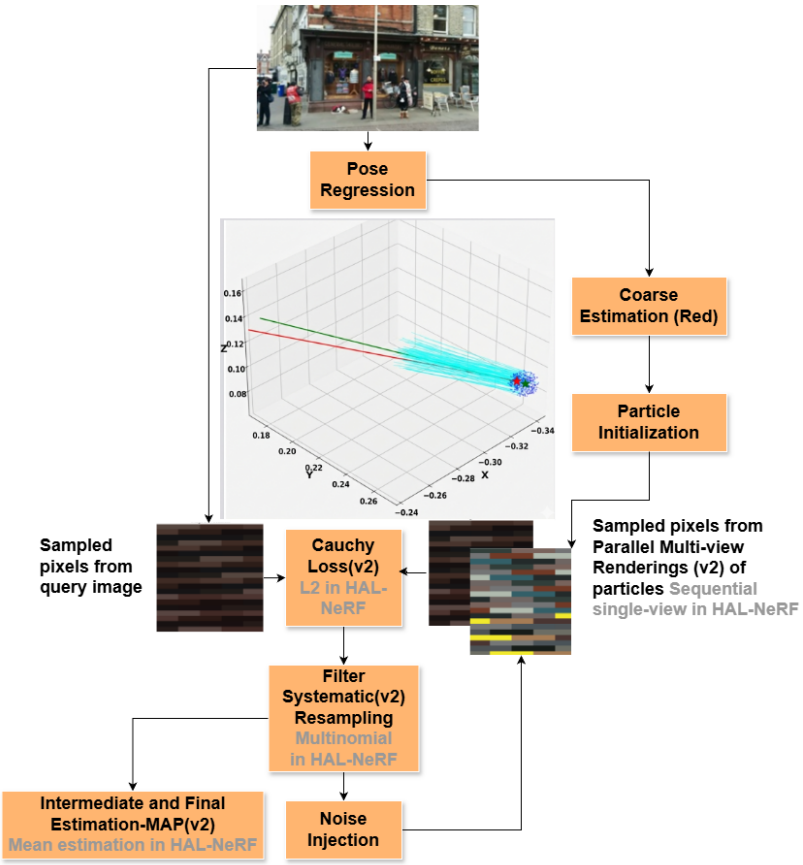


FIGURE 1: HAL-NeRF v2 Pipeline: Given a query image, a coarse pose estimate is first obtained through the pose regression module. Around this estimate, a set of particles (candidate poses) is initialized. For each particle, a set of random query-image pixels is compared against the corresponding pixels rendered by nerfacto, and a Cauchy loss is computed. This loss guides the resampling step, promoting particles that better explain the query image. After resampling, pose noise is injected into the surviving particles to encourage exploration. Loss evaluation, resampling, and noise injection is executed iteratively, while the coarse pose estimation is performed only once at the beginning, and the final pose is produced only after the last iteration. Where major updates are introduced in v2, gray annotations are used to show the corresponding HAL-NeRF components [19] for direct comparison.

3 intermediate layers and is trained with three losses: 1) L1 loss between poses matrices, 2) photometric loss between ground truth pose and NeRF-rendered pose, and 3)feature loss (intermediate representations are selected as features). The DFnet pipeline uses NeRFh representation in a data augmentation online (as training progresses) scheme that the DFnet author's named Random View Synthesis (RVS). RVS creates synthetic pairs of poses and images that are rendered from the NeRF given a pose that is not included in the dataset (perturbations around ground-truth poses). DFnet is used with minimal parameter changes in the "HAL-NeRF v2" pipeline. The key difference is that data augmentation (RVS) uses splatfacto renderings instead of NeRFh renderings. This is a factor that changed the performance of DFnet, as discussed in the IV Results section.

(a) Query image. (b) nerfacto rendering.

FIGURE 2: Query image with transient objects and nerfacto-rendered views.

C. POSE REFINEMENT

This section refers to the steps that follow the initial pose estimation until the final pose estimation, i.e., the initialization,resampling and update (noise injection) of the particle filter (Figure 1). These steps form an iterative process. The pose refinement method is applied at test time. For a useful

scenario, it should be close to instant. In HAL-NeRF [19], despite the accurate results, the process took 50 iterations, resulting in 26 seconds of processing per image. Processing was reduced to 6 seconds for 50 iterations by passing in parallel many rays from the same MLP. That was done sequentially by HAL-NeRF. In addition, the enhanced pipeline does not require 50 iterations. Therefore , with 10 iterations, the processing time is close to 1 s, achieving small errors. The refinements included:

1) Particle Filter Initialisation

The initialization of the particle filter is done by creating a number N of poses. These poses, which are then used to render novel views from the NeRF (NeRF Rendered images in Figure 1), are random but they are spread uniformly in positions and orientations around DFnet's estimation. The perturbed poses were constructed by applying translational and rotational offsets (px,py,pz,rx,ry,rz) to the coarse predicted pose. This defined particle dispersion for sampling alternative hypotheses. Therefore, in total, there are 6 parameters. These parameters should be configured according to the expected errors in the coarse estimate. Particle initialization in HAL-NeRF v2 is restricted to a compact region around the initial pose, spanning $\pm 4^\circ$ in rotation, ± 0.5 m in lateral and depth directions, and ± 0.05 m in height, ensuring efficient yet robust convergence during refinement. In Figure 3 (a), there is an example of particle initialization.

Let the predicted pose (from pose regressor) be

$$T_{\text{pred}} = \begin{bmatrix} R_{\text{pred}} & \mathbf{t}_{\text{pred}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in SE(3) \tag{1}$$

where $R_{\text{pred}} \in SO(3)$ is the rotation and $\mathbf{t}_{\text{pred}} \in \mathbb{R}^3$ is the translation. The initial particles position and rotation samples are:

$$\mathcal{P}_0 = \{\mathbf{t}_i, R_i\}_{i=1}^N \tag{2}$$

2) Particle Filter Weights calculation

To update and resample the particle filter, loss and weights are calculated per particle(pose). Instead of rendering full images, a NeRF model can output sampled pixels by its corresponding rays. This reduces time and memory demands by an important factor. For example, 200 candidate poses, rendering the whole images is heavy and time consuming. These sampled pixels are used to measure the photometric loss which in HAL-NeRF was calculated by the mean squared error (mse) between the query image's pixels and NeRF-rendered pixels. In this paper, mse is replaced by Cauchy loss to account for outliers caused by transient objects. Cauchy photometric loss is given by:

$$L_{\text{photo}} = \frac{1}{N} \sum_{i=1}^{N} \log \left(1 + \frac{(I_i^{\text{pred}} - I_i^{\text{gt}})^2}{\sigma^2} \right) \tag{3}$$

where L_{photo} is the average photometric loss over N sampled pixels or rays, I_i^{pred} is the predicted color of the i-th pixel/ray, I_i^{gt} is the corresponding ground-truth value, $r_i = I_i^{\text{pred}} - I_i^{\text{gt}}$ is the residual, and $\sigma > 0$ is a scale parameter controlling robustness to outliers. In HAL-NeRF v2 σ was chosen equal to 1, which is relatively moderate value for images scaled to floats in [0,1]. Unlike the L2 loss, which grows quadratically with the residual and therefore assigns disproportionate influence to large photometric discrepancies, the Cauchy loss grows logarithmically and exhibits a bounded influence function. This property makes it particularly suitable for localization in dynamic or partially reconstructed scenes, where transient objects (e.g., pedestrians) or missing geometry lead to localized but severe pixel-wise mismatches between the query image and NeRF renderings. In such cases, an L2-based likelihood can cause particle weights to collapse prematurely toward incorrect hypotheses. By contrast, the Cauchy loss downweights these outliers, allowing the particle filter to remain stable and preserve plausible pose hypotheses during refinement.

The Cauchy loss demonstrated better performance compared to L2 loss used in HAL-NeRF . This is attributed to the fact that it compensates for outliers caused by transient objects. Note that nerfacto effectively removes transient objects (Figure 2). Therefore, rendered images present mismatches with query images in regions where query images are covered by transient objects.

The number of pixels and the number of particles are parameters that define the balance between processing time and accuracy. By passing rays from multiple viewpoints through nerfacto, HAL-NeRF v2 effectively permits a higher number of sampled pixels and candidate poses and reduces processing time.

The loss is used to calculate the weights for each particle according to:

$$w_i = \left(\frac{1}{l_i} \right)^x, \tag{4}$$

where l_i is the loss of the i^{th} particle and x is a scaling exponent (set to $x = 8$ following the Loc-NeRF formulation). This parameter further enhances the contribution of particles that correspond to lower losses. These weights are then used in the particle filter update and resampling steps.

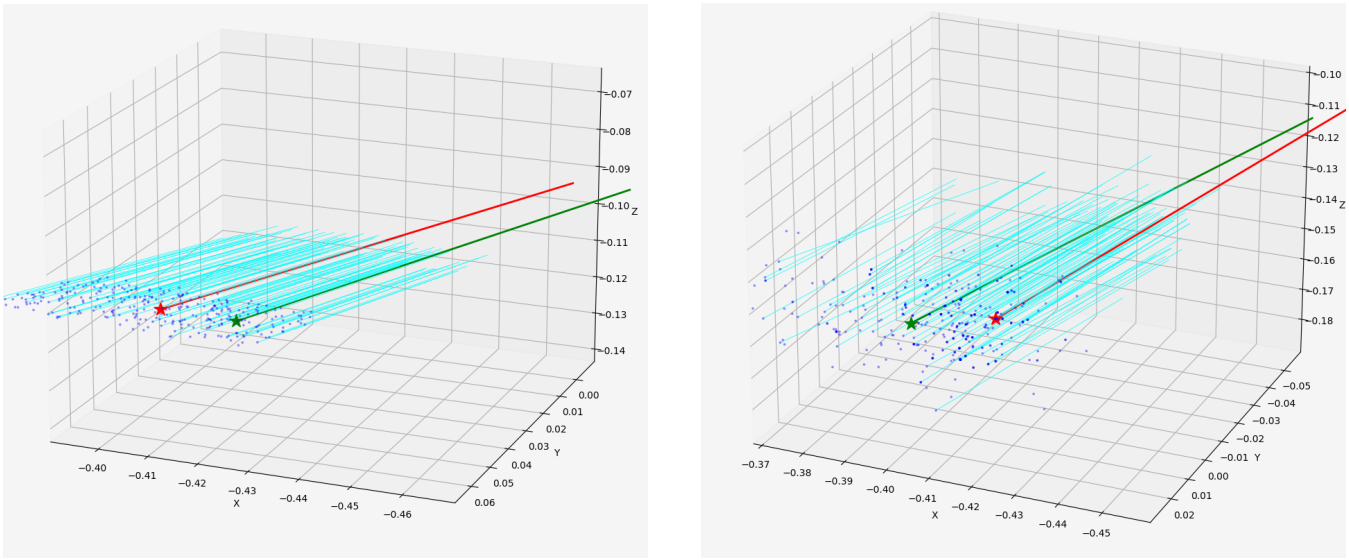
3) Filter Resampling

In the resampling stage, particles with higher likelihood weights are preferentially selected for propagation, thereby concentrating computational effort on the most promising pose hypotheses.

A common approach is *multinomial resampling*, in which new particle indices are drawn according to a categorical distribution proportional to their normalized weights:

$$i_k^* \sim \text{Categorical}\big(\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_N\big), \tag{5}$$

where each index i_k^* is drawn with probability $P(i_k^* = i) = \tilde{w}_i$. While simple, multinomial resampling exhibits high variance, as some particles may be overrepresented while others are lost prematurely, particularly when weight distributions are skewed.



(a) Particle initialization

(b) Particle second state.

FIGURE 3: Example of particle initialization and the second iteration of particle filtering. Candidate poses (cyan), ground-truth pose (green), and current estimate (red) are shown for both stages. Coordinate values are reported in the scene-specific normalized coordinate system used during NeRF training.

(a) Query image.

(b) nerfacto rendering.

FIGURE 4: Query image and NeRF and GS renderingswith different appearance.

In contrast, we employ *systematic resampling* , which stratifies the cumulative weight distribution to ensure more uniform particle selection. Specifically, a single random offset $u_1 \sim \mathcal{U}\big(0, \frac{1}{N}\big)$ is drawn, and the sampling positions are computed as

$$u_k = u_1 + \frac{k-1}{N}, \qquad k = 1, 2, \ldots, N. \tag{6}$$

Each new particle index i_k^* satisfies

$$i_k^* = \min\Big\{\, i \;\Big|\; \sum_{j=1}^{i} \tilde{w}_j \geq u_k \,\Big\}. \tag{7}$$

This procedure guarantees one sample per equal-probability interval, effectively reducing variance and leading to smoother convergence compared to multinomial resampling.

The resampled particle set is then

$$\mathcal{P}' = \big\{\, \mathbf{x}'_k = \mathbf{x}_{i_k^*} \,\big\}_{k=1}^{N}, \tag{8}$$

where each particle state $\mathbf{x}_i = \{\mathbf{t}_i, R_i\}$ consists of translation $\mathbf{t}_i \in \mathbb{R}^3$ and rotation $R_i \in \mathrm{SO}(3)$.

Finally, the updated hypothesis pool combines the resampled particles with the previous best estimate:

$$\mathcal{P} \leftarrow \big\{\, (R^\star, \mathbf{t}^\star) \,\big\} \cup \big\{\, (R_i, \mathbf{t}_i) \;\big|\; i = 1, \ldots, N-1 \big\}. \tag{9}$$

Empirically, systematic resampling produced a more consistent refinement behavior in HAL-NeRF v2, producing smoother convergence and reduced particle degeneracy compared to multinomial resampling.

4) Noise Injection

After resampling, the particles are perturbed to preserve diversity and prevent sample impoverishment. This *noise injection* step allows controlled exploration of the local pose space surrounding high-likelihood hypotheses.

Each particle state is represented as

$$\mathbf{x}_i = \{\, \mathbf{p}_i, R_i \,\}, \tag{10}$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the position and $R_i \in \mathrm{SO}(3)$ denotes its orientation.

a: Translational noise

In this work, translational noise is modeled using a zero-mean *uniform* distribution to produce an even spatial spread around each hypothesis:

$$\mathbf{p}'_i = \mathbf{p}_i + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}, \quad u_j \sim \mathcal{U}(-\Delta_{p_j}, \Delta_{p_j}), \quad j \in \{x, y, z\}. \tag{11}$$

The bounds Δ_{p_j} define the perturbation range along each spatial axis. Unlike Gaussian noise, uniform perturbation ensures that all local directions are sampled equally within a bounded

region, avoiding bias toward the origin and preventing large outliers.

b: Rotational noise

For orientations, we inject small random perturbations in the tangent space of the rotation group $\mathrm{SO}(3)$. Each new rotation is generated as

$$R_i' = \exp\big([\delta\boldsymbol{\theta}]_\times\big)R_i, \quad \delta\boldsymbol{\theta} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, \quad n_k \sim \mathcal{N}(0, \sigma_{r_k}^2), \tag{12}$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix operator, and the exponential map $\exp(\cdot)$ transforms the perturbation from the Lie algebra $\mathfrak{so}(3)$ to the manifold $\mathrm{SO}(3)$. This additive noise on the rotation vector corresponds to a small-angle random rotation around the current orientation, promoting smooth exploration on the rotational manifold.

Overall, the combination of bounded translational noise and Gaussian rotational noise provides a balanced trade-off between local exploration and convergence stability.

c: Parameterization

The six parameters $\{\sigma_{p_x}, \sigma_{p_y}, \sigma_{p_z}, \sigma_{r_x}, \sigma_{r_y}, \sigma_{r_z}\}$ govern the amount of translational and rotational diffusion. In the original Loc-NeRF implementation, these values were fixed a priori. In contrast, HAL-NeRF v2 adaptively scales these parameters according to the estimated uncertainty of the pose regressor. Noise magnitudes are gradually reduced after each iteration, allowing the filter to transition from exploration to convergence.

After perturbation, the updated particle set $\mathcal{P}' = \{\mathbf{x}_i'\}_{i=1}^N$ is used to generate new NeRF renderings for each hypothesized pose. These renderings are compared with the target images and the resulting losses are used to assign weights and refine the particle distribution.

5) Intermediate Pose Estimation per Iteration and Final Estimation

In the original HAL-NeRF framework, the intermediate and final pose estimates were computed as the weighted averages of the particle set. Given a set of N particles $\mathcal{P} = \{(\mathbf{p}_i, R_i, w_i)\}_{i=1}^N$, where w_i denotes the normalized particle weight, the estimated position and rotation were defined as:

$$\hat{\mathbf{p}}_{\text{HAL-NeRF}} = \sum_{i=1}^{N} w_i\, \mathbf{p}_i, \tag{13}$$

$$\hat{R}_{\text{HAL-NeRF}} = \mathrm{Exp}\left(\sum_{i=1}^{N} w_i\, \mathrm{Log}(R_i)\right), \tag{14}$$

where $\mathrm{Log}(\cdot)$ and $\mathrm{Exp}(\cdot)$ denote the logarithmic and exponential maps between the rotation group $\mathrm{SO}(3)$ and its tangent space $\mathfrak{so}(3)$. This formulation corresponds to a weighted mean estimate (Minimum Mean Square Error, MMSE) of the pose distribution.

In contrast, HAL-NeRF v2 adopts a maximum a posteriori (MAP) approach, selecting the particle with the highest likelihood as the best pose hypothesis. This modification reduces the effect of particle degeneracy and provides more stable pose estimates in cases of multi-modal likelihood distributions:

$$(\hat{\mathbf{p}}_{\text{HAL-NeRF v2}}, \hat{R}_{\text{HAL-NeRF v2}}) = \arg \max_{i \in \{1,\ldots,N\}} w_i. \tag{15}$$

In this setting, the intermediate pose estimate at each iteration corresponds to the pose of the highest-weight particle, and the final pose is taken as the best estimate from the last iteration of the filter.

This change from a weighted mean to a MAP estimation reflects the observation that the maximum-weight particle—representing the mode of the posterior distribution—provides more accurate and robust results, especially in non-Gaussian localization scenarios.

In Figure 3 (b), an example of the second state of the particle filter is presented. In this case , particles are drawn towards the ground truth-; however, the best estimate is placed on the opposite side of the ground truth.

IV. RESULTS

A. DATASET

The Cambridge Landmarks dataset [3] is referenced by visual localization methods, especially those that use NeRF representations. The data set consists of camera captures of Landmarks in Cambridge,UK. 2-D captures of specific camera trajectories are used to train a pose estimation system, and the remaining 2-D captures are reserved for testing. The data set can be characterized as an "in-the-wild" data set because transient objects (mainly pedestrians) and variations in the illumination are present in the scenes . The dataset scenes that are commonly used are Kings College, Old Hospital,Shop Facade and St Mary's Church, which consist of 1562 (343 test) images, 1078 (181 test), 330 (103 test) images and 2011 (530 test) images,respectively.

B. IMPLEMENTATION DETAILS

All experiments were conducted on a workstation equipped with a NVIDIA GeForce RTX 3060 GPU (12 GB VRAM) using CUDA 12.8. The system ran Ubuntu 22.04 LTS with PyTorch 2.5.0 and Python 3.10. Training and evaluation were implemented in PyTorch, leveraging GPU acceleration for both NeRF rendering and pose regression networks. Regarding particle and pixel numbers, 400 particles and 256 pixel samples were the option of choice. Generally, a balanced number of particles and pixels provided the best results, e.g. 200 particles with 512 pixel samples. The number of particles determines how many candidate poses can be explored during refinement, while the number of pixel samples controls the fairness and robustness of their photometric comparison. Due to GPU memory limitations, further increasing these parameters was not feasible. Although this constraint may

TABLE 1: **Synthetic data effects on pose regression in Cambridge Landmarks**

Methods	Kings	Hospital	Shop	Church	Average
DFnet [4]	0.73/2.37	2.00/2.98	0.67/2.21	1.37/4.03	1.19/2.90
Direct [4]	0.43/0.87	0.46/0.87	0.16/0.59	0.50/1.49	0.39/0.96
DFnet GS	0.71/2.38	1.94/2.41	0.54/2.21	1.31/2.93	1.12/2.48
Direct GS	0.42/0.84	0.46/0.84	0.16/0.54	0.44/1.33	0.37/0.82

Results are reported as translational(meters)/rotational errors(degrees). DFNet and Direct correspond to the original DFNet and its variant with direct feature matching, respectively. DFNet GS and Direct GS denote the same methods trained on synthetic data generated with Splatfacto.

slightly reduce accuracy, it ensures computational efficiency and keeps inference time manageable.

A difference from the DFnet method [4] is that the ground truth poses were extracted by COLMAP [28]. This was essential to train a photorealistic NeRF. COLMAP creates a set of camera extrinsics that are in a space of arbitrary scale per dimension. However, the correspondences between ground-truth that are in real-world dimensions and COLMAP were calculated and the pipeline's outcome , i.e. pose estimation, is translated to the real-world dimensions.

C. COMPARATIVE RESULTS OF SYNTHETIC DATA FOR POSE REGRESSION

In the setup of this method, the pose regressor is retrained with splatfacto views and then refined mainly to correct its position. This was the main adaptation. Concerning the pose regression module of the presented pipeline, using splatfacto instead of NeRFh was the major adaptation in the pose regression and was adequate to achieve slightly smaller translation and rotational errors in pose regression. Table 1 reports the results of DFnet and DFnetDM, each trained with either NeRFh or Splatfacto synthetic images. The models trained with Splatfacto data achieve lower rotational and translational errors, indicating superior convergence compared to those trained with NeRFh.

Nerfacto and splatfacto are up-to-date methods that incorporate best practices from different NeRF and GS publications. However, they do not leverage the appearance information to the point that NeRFh does. NeRFh (histogram) is an exposure-adaptive NeRF that is more suited than NeRFw to retain the right exposure of its camera. While NeRFh preserves appearance and exposure, it struggles with fine details. The DFnet authors argued that since NeRFh can reproduce the appearance of the query image and present higher PSNR than NeRFw, it is better suited for pose regression However, in practical scenarios, appearance and exposure are also dynamic. Therefore, it is debatable whether appearance should be a key factor in pose regression. Nerfacto and splatfacto are also equipped with appearance embedding. However, the appearance is not always exactly reproduced in the nerfacto/splatfacto renderings (Figure 4).

D. ABLATION FOR POSE REFINEMENT DESIGN OPTIONS

In Table 2, the effect of modifying pose refinement steps is examined. The table shows results if one of the design

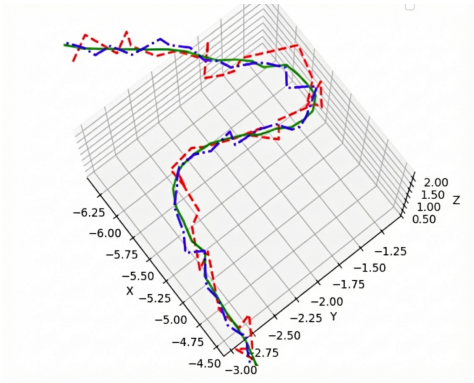


FIGURE 5: Ground truth trajectory (green line), coarse pose estimations(red line) and refined poses (blue line). Coordinate values are expressed in meters.

TABLE 2: **Results depending on pose refinement design choices after 10 iterations**

Methods	Kings	Hospital	Shop	Church	Average
First Est.	0.42/0.69	0.46/0.71	0.16/0.54	0.44/1.33	0.37/0.82
Pose Ref.	**0.08/0.61**	**0.16/0.63**	**0.04/0.49**	**0.07/0.71**	**0.09/0.61**
Multin. R	0.15/0.68	0.22/0.72	0.08/0.54	0.12/0.77	0.14/0.67
W. average	0.09/0.67	0.17/0.70	0.06/0.50	0.08/0.75	0.10/0.65
L2 loss	0.10/0.66	0.24/0.68	0.09/0.51	0.09/0.72	0.13/0.64

Results are reported as translational(meters)/rotational errors(degrees). Pose Ref.: proposed pose refinement, Multin. R: multinomial resampling instead of systematic resampling, W. average: weighted average estimation instead of maximum probability estimation, L2 loss: L2 loss instead of Cauchy Loss

options is replaced by the original pose refinement step i.e. multinomial resampling in place of systematic resampling, weighted average estimation in place of MAP and mse loss in place of Cauchy loss.

The greater improvement was due to systematic resampling. Systematic resampling keeps more candidate poses with high-probability in the particle filter, instead of keeping only the highest probabilities. In this way a more diverse set of poses can be examined in a next iteration. MAP estimation avoids noisy particles that may have been kept in the filter, and Cauchy loss compensates for pixel intensities mismatches due to transient effects. As shown in Table 2, replacing the Cauchy loss with an L2 loss consistently degrades translational accuracy across all scenes, confirming that robust photometric modelling is critical for stable pose refinement in the presence of transient effects.

Note that design choices are interdependent and other combinations could work as they did in HAL-NeRF v1. In addition, in Figure 5 which shows a segment of a camera trajectory along with position estimations, most poses get their positions closer to the ground-truth, due to refinement. However, there are cases which show the limitations of the proposed pipeline in converging fast to low position errors.

E. COMPARATIVE RESULTS WITH OTHER METHODS

In Table 3, the comparative results with other state-of-the-art methods are presented. HAL-NeRF v2 presents lower trans-

TABLE 3: **Comparison with other methods in Cambridge Landmarks**

Methods	Kings	Hospital	Shop	Church	Average
DFnetdm [4]	0.43/0.87	0.46/0.87	0.16/0.59	0.50/1.49	0.38/0.96
Nefes [15]	0.37/0.54	0.52/0.88	0.15/0.53	0.37/1.14	0.35/0.77
Crossfire [26]	0.47/0.7	0.43/0.7	0.20/1.2	0.39/1.4	0.37/1.0
NeRFMatch [16]	0.12/**0.2**	0.21/**0.4**	0.09/**0.4**	0.11/**0.4**	0.11/**0.3**
HAL-NeRF v2	**0.08**/0.61	**0.16**/0.63	**0.04**/0.49	**0.07**/0.71	**0.09**/0.61

Results are reported as translational(meters)/rotational errors(degrees).

lation and rotational errors than the best performing methods. Its main advantage is the fast exploration of the NeRF space to refine the coarse pose of APR methods. However, it can refine pose up to scale due to the NeRF rendering ambiguity. Specifically, NeRF renderings from distances up to 20 cm may differ very little.

Recent works such as NeRFect Match [16] leverage NeRF's internal volumetric features for explicit 2D–3D matching, achieving competitive localization accuracy on Cambridge Landmarks at the cost of increased computational complexity due to transformer-based matching and iterative NeRF rendering. In contrast, HAL-NeRF focuses on integrating NeRF-derived supervision into a compact regression-and-refinement pipeline. By jointly processing multiple viewpoints per iteration and employing lightweight robust loss formulations, our method achieves similar accuracy with significantly reduced runtime and improved convergence stability.

V. DISCUSSION

This paper presents an extension to the HAL-NeRF method, achieving significantly faster low positional error and rotational errors for challenging scenes of Cambridge Landmarks dataset that were trained with few images that present transient objects. The HAL-NeRF v2 re-examines the combination of a pose regression module with a pose refinement module by focusing on the steps of particle filter steps.

For pose regression, the only major difference was the use of splatfacto renderings for data augmentation. This allowed for the fast creation of more synthetic data, reducing the training time. Compared to other implicit representations (e.g. NeRFh), splatfacto renderings were more efficient in preserving fine details, which led to slightly lower errors in the output of the pose regressor. However, the appearance was not preserved with the same efficiency as NeRFh, which may be the reason for its limitation in accuracy. An implicit representation, that combines sharp detail renderings and tunes the appearance of synthetic images, might lead to even lower errors, in the pose regressor.

Both HAL-NeRF and HAL-NeRF v2 used DFnet as pose regressor. Although DFnet introduced many effective intuitive and localization specific processing ideas to adopt, its backbone, i.e. VGG16, was the only one used in HAL-NeRF v2. Other CNN architectures are considered more effective in down-stream tasks and could be used in the HAL-NeRF pose regressor. Similarly to DFNet, we employ conservative data augmentation during coarse pose regression via novel view synthesis, while avoiding large-scale synthetic supervision during fine pose refinement due to potential rendering inaccuracies.

In contrast, the pose refinement stage underwent substantial improvement. Four modifications—(a) Cauchy loss, (b) systematic resampling, (c) parallel multi-view evaluation, and (d) maximum-likelihood estimation—significantly accelerated convergence . Crucially, the refinement stage consistently improved performance across scenes with diverse challenges: for example, the Shop scene contains transient objects, while the Hospital scene exhibits partial 3D reconstruction. In all cases, the proposed refinement method outperformed its variants.

The efficiency of the refinement module, both in accuracy and runtime, depends on the quality of the coarse pose estimation. In our experiments, the search space was constrained, given that low rotational error is expected, which is a common trait among state-of-the-art regressors. This constraint allowed particles to explore a wider region in translation while restricting orientation hypotheses. In practice, particle initialization should be chosen based on the expected or previously reported error for each scene, but the refinement module remains broadly useful even when such errors vary.

If computational resources allow, increasing the number of particles can further reduce errors, especially when the coarse estimate is less accurate. Besides, additional refinement iterations can improve precision, albeit at an increased computational cost.

The challenge for NeRF-based localization is to be able to operate in dynamic scenes. Cambridge Landmarks is a dynamic scene, in the sense that pedestrians move along the 2-D captures. However, they cover only a small portion of the image. HAL-NeRF v2 should be tested in more dynamic scene datasets, to estimate its robustness against larger changes in scene, but to the best of our knowledge, there is no other available dataset to benchmark visual localization in dynamic scenes.

VI. CONCLUSION

HAL-NeRF v2 introduces a fast and robust localization pipeline that unites pose regression with Monte Carlo refinement using NeRF and Gaussian Splatting representations. Using robust losses, efficient resampling, and MAP estimation, it maintains the precision of HAL-NeRF v1 while improving runtime. Future work will explore the adaptation to more dynamic scenes.

REFERENCES

[1] H. Lim, S. N. Sinha, M. F. Cohen, M. Uyttendaele, and H. J. Kim, "Real-time monocular image-based 6-DoF localization," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 476–492, 2015.

[2] T. Wen, K. Jiang, B. Wijaya, H. Li, M. Yang, and D. Yang, "TM3Loc: Tightly-coupled monocular map matching for high precision vehicle localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20268–20281, 2022.

[3] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DoF camera relocalization," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 2938–2946, 2015.

[4] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu, "DFNet: Enhance Absolute Pose Regression with Direct Feature Matching," in *European Conference on Computer Vision (ECCV)*, pp. 1–17, 2022.

[5] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization,"in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 4762–4769, 2016.

[6] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 5974–5983, 2017.

[7] Y. Shavit, R. Ferens, and Y. Keller, "Paying attention to activation maps in camera pose regression," *arXiv preprint arXiv:2103.11477*, 2021.

[8] J. Wu, L. Ma, and X. Hu, "Delving deeper into convolutional neural networks for camera relocalization," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 5644–5651, 2017.

[9] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based Localization Using LSTMs for Structured Feature Correlation," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 627–637, 2017.

[10] Y. Shavit, R. Ferens, and Y. Keller, "Learning Multi-scene Absolute Pose Regression with Transformers," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pp. 2733–2742, 2021.

[11] Y. Shavit and Y. Keller, "Camera Pose Auto-Encoders for Improving Pose Regression," in *European Conference on Computer Vision (ECCV)*, pp. 140–157, 2022.

[12] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF: Inverting Neural Radiance Fields for Pose Estimation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 1323–1330, 2021.

[13] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-NeRF: Monte Carlo Localization using Neural Radiance Fields," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 4018–4025, 2023.

[14] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "LENS: Localization Enhanced by NeRF Synthesis," in *Proc. Conf. on Robot Learning (CoRL)*, pp. 1347–1356, 2022.

[15] S. Chen, Y. Bhalgat, X. Li, J.-W. Bian, K. Li, Z. Wang, and V. A. Prisacariu, "Neural Refinement for Absolute Pose Regression with Feature Synthesis," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 20987–20996, 2024.

[16] Q. Zhou, M. Maximov, O. Litany, and L. Leal-Taixé, "The NeRFect Match: Exploring NeRF Features for Visual Localization," *arXiv preprint arXiv:2403.09577*, 2024.

[17] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Trans. Graph.*, vol. 42, no. 4, Jul. 2023. DOI: 10.1145/3592433.

[19] A. Reppas, G.-A. Cheimariotis, P. K. Papadopoulos, P. Frasiolas, and D. Zarpalas, "HAL-NeRF: High Accuracy Localization Leveraging Neural Radiance Fields," *Proc. IEEE Int. Conf. on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, Lisbon, Portugal, 2025, pp. 117–124, doi: 10.1109/AIxVR63409.2025.00024.

[20] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, 2012.

[21] S. Tang, S. Tang, A. Tagliasacchi, P. Tan, and Y. Furukawa, "Neumap: Neural coordinate mapping by auto-transdecoder for camera localization," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 929–939, 2023.

[22] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 7210–7219, 2021.

[23] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, and others, "Nerfstudio: A Modular Framework for Neural Radiance Field Development," in *ACM SIGGRAPH Conference Proceedings*, pp. 1–12, 2023.

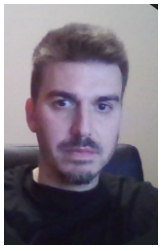
[24] Z. Zhu, Y. Chen, Z. Wu, C. Hou, Y. Shi, C. Li, P. Li, H. Zhao, and G. Zhou, "LATITUDE: Robotic Global Localization with Truncated Dynamic Low-Pass Filter in City-scale NeRF," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 8326–8332, 2023.

[25] J. Lin, G. Gu, B. Wu, L. Fan, R. Chen, L. Liu, and J. Ye, "Learning neural volumetric pose features for camera localization," in Proc. ECCV, Cham, Switzerland: Springer Nature, Sep. 2024, pp. 198–214.

[26] A. Moreau, N. Piasco, M. Bennehar, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Crossfire: Camera Relocalization on Self-supervised Features from an Implicit Representation," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pp. 252–262, 2023.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[28] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016.



GRIGORIOS-ARIS CHEIMARIOTIS received his degree in Electrical and Computer Engineering from AUTH in 2010 and a Master's in Medical Informatics from AUTH in 2013. Since 2014, he has worked at AUTH, INAB/CERTH, and DUTH, and since 2024 he has been with ITI, contributing to European and national research projects. He obtained his PhD from AUTH in 2020, with a dissertation focused on Medical Image Analysis. He has (co-)authored 10 publications in international scientific journals and 19 presentations at international conferences.

DIMITRIOS ZARPALAS Dimitrios Zarpalas is a member of Information Technologies Institute since 2007. He holds the diploma of Electrical and Computer Engineer from Aristotle University of Thessaloniki, A.U.Th, an MSc in Electrical Engineering (focusing on computer vision) from The Pennsylvania State University, and a PhD in medical informatics (Health Science School, department of Medicine, A.U.Th). His main research interest are on 3D/4D computer vision and machine learning, such as tele-immersion applications: 4D reconstruction of moving humans, their "hologram" compression and transmission in real-time; 3D motion capturing, analysis and evaluation; 3D object recognition and 3D shape descriptor extraction; 3D medical image processing, shape analysis of anatomical structures; while in the past has also worked in indexing, search and retrieval and classification of 3D objects, proteins and 3D model watermarking. He has (co-)authored more than 75 papers in peer reviewed international journals, conference proceedings, and books (including one IEEE Distinguished paper and one IEEE conference best paper award).