



Delft University of Technology

CoPR: Toward Accurate Visual Localization With Continuous Place-Descriptor Regression

Zaffar, M.; Nan, L.; Kooij, J. F. P.

DOI

[10.1109/TRO.2023.3262106](https://doi.org/10.1109/TRO.2023.3262106)

Publication date

2023

Document Version

Final published version

Published in

IEEE Transactions on Robotics

Citation (APA)

Zaffar, M., Nan, L., & Kooij, J. F. P. (2023). CoPR: Toward Accurate Visual Localization With Continuous Place-Descriptor Regression. *IEEE Transactions on Robotics*, 39(4), 2825-2841. <https://doi.org/10.1109/TRO.2023.3262106>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

CoPR: Toward Accurate Visual Localization With Continuous Place-Descriptor Regression

Mubariz Zaffar , Liangliang Nan , and Julian Francisco Pieter Kooij 

Abstract—Visual place recognition (VPR) is an image-based localization method that estimates the camera location of a query image by retrieving the most similar reference image from a map of geo-tagged reference images. In this work, we look into two fundamental bottlenecks for its localization accuracy: 1) reference map sparseness and 2) viewpoint invariance. First, the reference images for VPR are only available at sparse poses in a map, which enforces an upper bound on the maximum achievable localization accuracy through VPR. We, therefore, propose Continuous Place-descriptor Regression (CoPR) to densify the map and improve localization accuracy. We study various interpolation and extrapolation models to regress additional VPR feature descriptors from only the existing references. Second, we compare different feature encoders and show that CoPR presents value for all of them. We evaluate our models on three existing public datasets and report on average around 30% improvement in VPR-based localization accuracy using CoPR, on top of the 15% increase by using a viewpoint-variant loss for the feature encoder. The complementary relation between CoPR and relative pose estimation is also discussed.

Index Terms—Continuous Place-descriptor Regression (CoPR), pose estimation, visual localization (VL), visual place recognition (VPR).

I. INTRODUCTION

ONE of the key research problems for robotics and computer vision is accurate visual localization (VL), i.e., to localize a robot in a map using as input only an image from the robot’s camera [1]. Various parallel research directions have emerged within VL. A top-level distinction can be made between purely image-based approaches and 3-D-structure-based approaches. The former ones are simple and efficient but have lower localization accuracy while the latter ones are more accurate at the cost of increased computation complexity and maintenance effort [2]. Purely image-based approaches could be further divided into visual place recognition (VPR) [3],

absolute pose regression (APR) [4], and relative pose estimation (RPE) [5]. Given their efficiency and scalability, VPR techniques are often used in robotics for loop closure detection or 3-D reconstruction. However, improving their performance remains an ongoing research challenge [6], [7].

In VPR, the task is to find for a query image the best matching reference image from a set of prerecorded geo-tagged reference images (i.e., the reference map) [8]. Each reference image is considered a “place,” and the geo-location of the best-matched reference is then the estimated location (“place”) of the query image. Whereas VPR relies on image retrieval, in APR, a neural network directly regresses the global coordinates for a query image, and the map is implicitly represented by the network weights. However, such APR methods do not generalize across viewpoints, as has been studied by Sattler et al. [9]. RPE on the other hand operates on two images with assumed nearby viewpoints and estimates from the overlapping image contents the relative translation and orientation between their corresponding camera coordinate frames. Since VPR performs coarse global localization, and RPE performs fine-grained localization by assuming coarse localization is solved, both techniques are often combined in the multistage approach, referred to as Coarse-to-Fine localization (CtF) [5], [10], [11]. RPE is, therefore, not an alternative to VPR, but a refinement step that is only successful if VPR was able to retrieve a nearby reference.

VPR remains less accurate than structure-based and CtF approaches [12], with a crucial reason being the discrete nature of the reference map in VPR. When a query image appears between two anchor locations in the reference map, a VPR system could at best only match this to the nearest spatial anchor location, incurring some minimal Euclidean distance error. This can become worse when query images and existing reference images span the same area but at offsets of parallel lines, as shown in Fig. 1. Therefore, we seek to add more references to the map (such as the blue poses in Fig. 1), a notion referred to as *map densification*. A trivial but often impractical solution to densification is by collecting more reference images. Alternatively, densification could be achieved by creating a 3-D model of the environment and rendering images at novel poses. However, creating and maintaining up-to-date 3-D models is computationally and storagewise expensive, and the resulting images are not photo-realistic [9], [13].

Since the VPR reference maps comprise compact feature descriptors of images, we suggest performing map densification in the *feature space* rather than the image space. We propose Continuous Place-descriptor Regression (CoPR) in feature space for

Manuscript received 27 May 2022; revised 18 November 2022; accepted 14 February 2023. Date of publication 12 April 2023; date of current version 8 August 2023. This work was supported by the 3D Urban Understanding (3DUU) lab funded by the TU Delft AI Initiative. This paper was recommended for publication by Associate Editor M. Chli and Editor F. Chaumette upon evaluation of the reviewers’ comments. (*Corresponding author: Mubariz Zaffar.*)

Mubariz Zaffar and Julian Francisco Pieter Kooij are with the Intelligent Vehicles Group, Department of Cognitive Robotics, Delft University of Technology (TU Delft), 2628 CD Delft, The Netherlands (e-mail: m.zaffar@tudelft.nl; J.F.P.Kooij@tudelft.nl).

Liangliang Nan is with the 3D Geoinformation Group, Faculty of Architecture and the Built Environment, Delft University of Technology (TU Delft), 2628 BL Delft, The Netherlands (e-mail: liangliang.nan@gmail.com).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TRO.2023.3262106>.

Digital Object Identifier 10.1109/TRO.2023.3262106

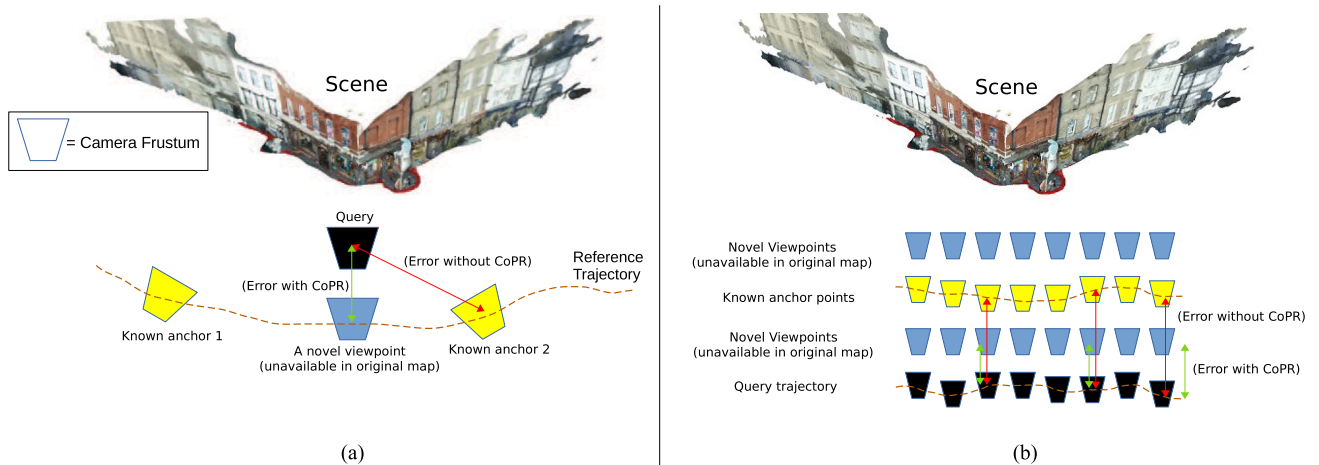


Fig. 1. Discrete treatment of VPR that leads to lower localization accuracy. Provided that only the *yellow* anchor reference poses are available in the map, the *black* query images could only be matched as close as possible to the base error. Regressing descriptors for the *blue* target viewpoints using interpolation or extrapolation given anchor reference descriptors could lead to improved localization accuracy for query images in VPR and, thus, reduce the base error. The scene shown in this figure is taken from the work of Sattler et al. [9].

VPR map densification.¹ Since, in CtF, the RPE step assumes the initial VPR step was performed correctly, we note that improving VPR could also address CtF errors that cannot be corrected by RPE, as we will also show in this work.

We argue for two requirements to benefit from such map densification: 1) a method of regressing meaningful feature descriptors for VPR at novel target viewpoints given anchor point feature descriptors, 2) an image-retrieval system that is viewpoint-variant and, therefore, could utilize the regressed descriptors at target viewpoints. Furthermore, the model for descriptor regression should only need existing anchor descriptors and relative poses between anchor locations and target viewpoints, at its input, and it should not require images of the scene from target viewpoints or expensive scene reconstruction [9].

To study the problem of descriptor regression, we further consider two possible schemes: 1) interpolation and 2) extrapolation. Both of these are relevant for map densification, where *interpolation* [see Fig. 1(a)] refers to interpolating to an intermediate location between some anchor points on the reference trajectory, while *extrapolation* [see Fig. 1(b)] refers to regressing descriptors around a given anchor reference pose. Since interpolation could even be performed using averaging of the nearest anchor points along the trajectory, i.e., by simply following the trend in the local feature space, we expect it to be an easier problem to solve than extrapolation. Extrapolation, on the other hand, is a more important requirement for map densification, because it enables us to potentially regress descriptors at or close to the query. Interpolation can at best only densify within an existing reference trajectory.

Finally, for a VPR system to benefit from map densification, it needs to retrieve the Euclidean closest match in the physical space as the best match in the feature space. This is

not enforced in VPR techniques trained with triplet-loss [3], classification-loss [14], and ranking-based-loss [15], where the correct/incorrect ground-truth (GT) match is discrete (leading to viewpoint invariance), instead of the continuous GT in distance-based loss [16]. If a VPR technique is viewpoint-invariant, both the blue trajectories in Fig. 1(b) would be incorrectly considered equally valid. Thus, we hypothesize that map densification and viewpoint variance should work hand in hand to make VPR-based localization more accurate. We show that a highly viewpoint-variant VPR technique in a densified reference map leads to the highest localization accuracy, among all the combinations originating from the different feature encoders and levels of map densification.

In summary, our contributions are as follows.

- 1) We investigate CoPR to densify a sparse VPR map through either interpolation or extrapolation of the feature descriptors to target poses, without requiring any new measurements (i.e., reference images).
- 2) We propose linear-regression-based techniques and a non-linear deep neural network for map densification and demonstrate the improvement in localization accuracy on three existing public datasets.
- 3) We report that different feature encoders can benefit from map densification and the best performance is achieved by using the most viewpoint-variant descriptors in a densified map.
- 4) We discuss the VPR failure cases where RPE cannot recover the correct pose without CoPR, highlighting the complementarity of these approaches for improving VL accuracy. We demonstrate the existence of such cases with real-world data.

II. RELATED WORK

In this section, we expand on the existing body of literature for VL, as reviewed in [2]: A system that consists of retrieving the pose (position + orientation) of a visual query material within a

¹This discrete nature of the reference map is also problematic for APR as reported in [9]. We hypothesize that APR could also benefit from map densification via descriptor regression, but this aspect is not explored in this work and we limit its scope to VPR.

known space representation. Such systems are further classified into direct and indirect methods. The direct methods consist of APR, structure-based localization, and CtF. The indirect methods are VPR approaches, which is a robotics problem, and image retrieval, which is a computer vision problem. Both of these mostly represent the same formulation but with a few differences regarding evaluation metrics and experimental setup as discussed in [7]. In this research, our scope is limited to VPR-based localization and its limitations, however, to understand these limitations and due to the significant overlap between various fields of VL and the collective benefit from map densification, we expand on all these fields in the following.

Structure-based approaches: These approaches use 2-D–3-D matching given 2-D pixels and 3-D scene coordinates to yield highly accurate pose estimates. Recent benchmarks [9], [12], [17] have shown that such structure-based approaches are state-of-the-art when it comes to accurate localization. The work of Li et al. [18] is seminal in this field that shows large-scale structure-based localization by proposing a co-occurrence prior to RANSAC and bidirectional matching of image features with 3-D points. Efficiency is of importance and Liu et al. [19] propose the use of global contextual information derived from the covisibility of 3-D points for 2-D–3-D matching. InLoc [20] presents a formulation for structure-based localization in indoor environments by using dense feature matching for textureless indoor scenes and view synthesis for verification. ActiveSearch [21] uses 2-D-to-3-D and 3-D-to-2-D matching for pose regression and candidate filtering while DSAC++ [22] uses learned scene-coordinate regression building upon DSAC [23]. Both of these techniques form the state of the art for structure-based 6-DoF camera localization [9]. While structure-based approaches are highly accurate, they require significant computations and have limited scalability, and maintaining and updating the corresponding potentially large-scale 3-D models is challenging.

Absolute Pose Regression: APR started from the seminal works of PoseNet [4] and the incremental build-up by the authors in [24] and [25] and has since seen many different variants of it e.g., the works in [26], [27], [28], and [29]. The objective of APR approaches is to memorize an environment given a set of images and their corresponding GT poses, such that given a new image, the network can generalize from the poses seen at training time and directly regress the new pose. In [30], an encoder–decoder architecture is employed with a final regressor network to regress the camera pose. Radwan et al. [31] present a multitask learning framework for visual-semantic APR and odometry. While APR methods are simple and efficient, they have been shown to suffer from degeneralization across viewpoints and appearances, and are unable to extrapolate to parallel trajectories [9].

Coarse-to-Fine localization: Another approach to the problem of accurate localization is a two-staged coarse-to-fine formulation, where the first stage is VPR and the second stage is RPE. This need for CtF approaches arises because the query trajectories and reference trajectories are usually far apart, and the coarse VPR stage can only at best retrieve the closest pose on the reference trajectory. Thus, there is always a base error in the coarse VPR stage, which is then reduced by the RPE

module for fine-grained localization. Laskar et al. [5] propose a CtF approach by using a Siamese network architecture for RPE. RelocNet [10] uses camera frustum overlap information at training time while CamNet [11] models the CtF localization approach in three separate modules with increasing fineness. The work in [32] models pose estimation by discovering and computing relative poses between predefined anchor locations in the map. Most of these CtF approaches model RPE as a pose regression problem given global descriptors leading to a lack of scene generalization. Thus, Sarlin et al. in PixLoc [33] instead learn local features useful for geometric 2-D-to-3-D matching, which can generalize to new scenes. SANet [34] also models the CtF localization pipeline using 2-D-to-3-D matching by learning scene coordinate regression and generalizes to new scenes. However, both of these approaches require a coarse 3-D model of the environment at their inputs.

VPR and image retrieval: VPR and image retrieval in essence represent the same problem: i.e., given a query image and a map of reference images, retrieve the nearest neighbor (NN) reference matches for that query image. Depending on whether the closest match is required (VPR) or all of the possible matches need to be retrieved (image retrieval), the problem favors loop-closure or 3-D modeling, as discussed in [7]. In this work, we use the two terminologies interchangeably to refer to the same problem. Both these tasks are usually treated as viewpoint-invariant and trained with losses such as triplet-loss [3], [35], [36], classification-loss [14], and ranking-based-loss [15]. These losses aim to align the feature representation for viewpoint-varied images of the same place, which explicitly favors viewpoint-invariance. On the other hand, more recent distance-based loss functions explicitly force the network to encode geometric information within the feature descriptors, such that the top-most retrieved images are also the geometrically-closest images [16], [37]. For our work, such a distance-based loss is highly relevant, since map densification could offer more benefit to VPR-based localization using viewpoint-variant feature descriptors than viewpoint-invariant descriptors.

Before dedicated datasets were developed for VPR, off-the-shelf convolutional neural network (CNN) features were utilized, Chen et al. [38] used features from the Overfeat Network [39] and combined them with the spatial filtering scheme of Seq-SLAM. The use of off-the-shelf features of AlexNet trained on ImageNet for VPR was studied by Sunderhauf et al. [40], who found that some layers were most robust to conditional variations than others. Chen et al. [14] proposed two neural networks, namely AMOSNet and HybridNet, which were trained specifically for VPR on the Specific Places Dataset (SPED).

Recently, contrastive learning has been the dominant trend in VPR, as shown in [3] and [36], which classifies a place as the same or different in a hard (0/1) manner, i.e., an image is considered as either the same place or a different place. But with multiple viewpoint-varied images of the same place, such a hard distinction is not possible and a soft distinction is required. For this purpose, Leyva-Vallina et al. in [41] present the concept of generalized contrastive loss based on image-content overlap. Previously discussed distance-based loss functions can also be classified as soft losses since they can distinguish between

multiple viewpoint-varied images of the same place. Other than this, VPR literature includes the use of ensembles of VPR techniques to reject false positives [42], [43].

Implicit scene representations: In addition to the concept of explicit 3-D models for structure-based approaches, implicit scene representation has been more popular recently, where the structure is stored within the parameters of a neural network. Such implicit scene representation could come from neural implicit representations [44], [45], differentiable volumetric rendering [46] or the more recent trends in Neural Radiance Fields (NeRF) [47]. If the structure is known, whether implicitly or explicitly, it is possible to synthesize images at new viewpoints of the scene. These synthesized images could be directly used for map densification in a VPR-based localization system [48], for pose verification in a CtF system [49] or for creating more training data for APR approaches [13]. Yen-Chen et al. [50] invert the NeRF process to refine the camera pose estimate given an initial coarse estimate. However, implicit scene representation approaches offer similar challenges as structure-based approaches for localization regarding maintaining and updating the scene representations. They also suffer from scalability and artifacts created in the image space, as reported in [13].

In summary, VPR is an efficient and easy-to-maintain localization method compared to structure-based approaches, it is more generalizable than APR techniques and simpler than multistaged CtF approaches; however, it remains less accurate than CtF and structure-based approaches, where this accuracy is related to the sparseness of the reference map at creation and viewpoint-variance of the feature encoder. One possibility to increase this localization accuracy as surveyed here is to use the CtF approaches in a retrieval-followed-by-regression manner; however, this itself depends on the quality of the initial coarse retrieval stage, i.e., VPR, such that an incorrectly retrieved coarse estimate leads to a definite failure of the complete CtF pipeline.

Therefore, we instead look in a different direction than CtF and explore some of the fundamental reasons for the inaccuracy of VPR. We investigate whether it is possible to increase VPR-based localization accuracy even without relying on RPE as a second stage and without requiring any additional measurements of the scene. For this, we look into densifying the map of descriptors and the benefits of such map densification for different types of VPR feature encoders.

III. METHODOLOGY

In this section, we first provide an overview of our problem statement. We then dedicate sections to introduce the concept of map densification (CoPR), the descriptor regression strategies for CoPR, and the different feature encoders for VPR. Finally, we discuss the relationship between CoPR and RPE.

A. Problem Statement

Given a set of reference images with known poses, VPR constructs a map $M = (R, P)$, where R is a set of reference descriptors, such that $f_i \in R$ is an N -dimensional feature descriptor with a corresponding pose $p_i \in P$. Each feature descriptor $f_i = G(I_i)$ is obtained from a reference image I_i using

an already trained and fixed feature extractor G , typically a neural network. The pose p_i is a 6-degree-of-freedom pose that specifies the location as a translation vector $t_i = (x, y, z)$, and a quaternion vector o_i specifying the 3-D orientation.

At test time, the objective is to find the pose p_q of a query image I_q , for which the query descriptor $f_q = G(I_q)$ is computed. The descriptor f_q is matched to all the reference descriptors in the set R , and the NN match $r_{nn} = \operatorname{argmin}_{r \in R} \|f_r - f_q\|_2$ is retrieved. The pose of the query image is then considered the same as that of the retrieved reference descriptor, i.e., $p_q = p_{nn}$. Ideally, the feature descriptors are constructed such that the resulting Euclidean translation error $e = \|t_q - t_{nn}\|_2$ is minimal. Hence, the assumption $p_q = p_{nn}$ is essentially an approximation $p_q \approx p_{nn}$ and would only be true in the unlikely event that the query is collected at the same pose as that of the retrieved reference in the map. Thus, the expected error $\mathbb{E}[e]$ is a nonzero *base error* of a VPR system. This base error is directly affected by the sparseness in the reference map: The further apart the reference samples are, the higher the base error could be.² Therefore, this work proposes to apply map densification for VPR as shown in Fig. 1.

B. Map Densification

To reduce the base error, we seek to extend the number of descriptors and poses in a given sparse map M_{sparse} . Since collecting more reference images is not always possible, we aim to perform densification using only existing reference descriptors in M_{sparse} without the need to collect more images at novel viewpoints. Such densification in feature space also has computational benefits since image-description is more computationally expensive than descriptor-regression, as shown in Section IV-H. Concretely, we propose to densify a sparse map $M_{\text{sparse}} = (R, P)$ by defining a set of target poses P' for which the corresponding descriptors R' are predicted via CoPR using one or more existing reference descriptors in R , which we will refer to as *anchor descriptors*. The resulting densified map $M_{\text{dense}} = (R \cup R', P \cup P')$ thus extends the original map M_{sparse} with the newly regressed target references.

Different strategies could be employed to define (a) which set of target poses P' to regress to, and (b) how to regress the descriptors for a target pose using the available anchor descriptors. We here explore two specific strategies for defining the set P' , namely 1) interpolating between the anchor points on the reference trajectory, and 2) extrapolating to nearby poses of an anchor pose that do not necessarily lie along the reference trajectory. Regression approaches will be discussed later in Section III-C.

The *interpolation scheme* assumes that the references in the sparse map are obtained in a sequence. Additional poses P' can be selected along the trajectory in between the poses available in P . Hence, any two subsequent references $a1 \in R$ and $a2 \in R$ can be selected as anchors, and one or more new target poses

²Clearly, if the query images appear at the exact same spot as that of the reference trajectory, map densification would not help. This, however, is highly unlikely and unrealistic in real-world situations as evident in existing VPR datasets [7].

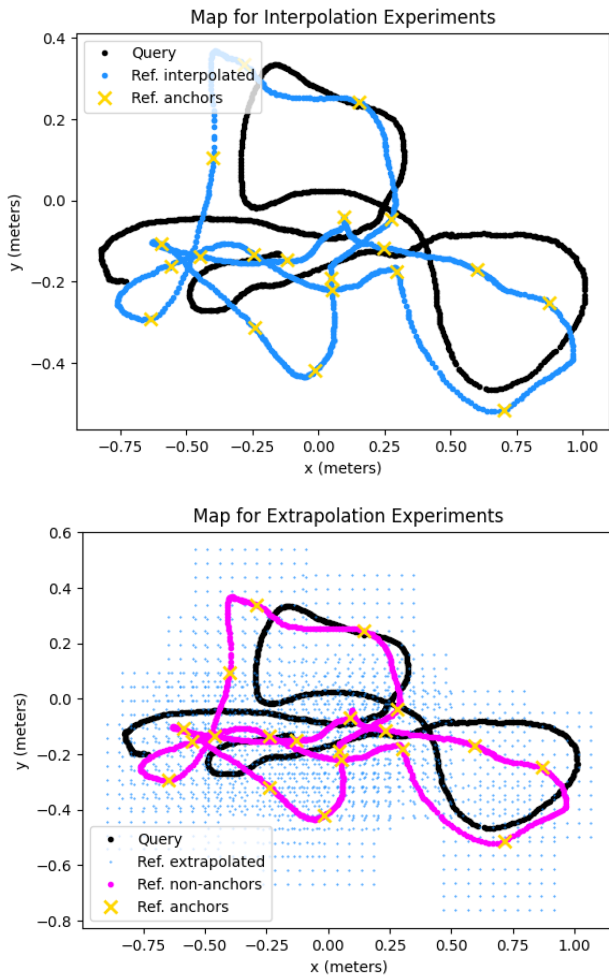


Fig. 2. Test setup for the interpolation and extrapolation experiments on the Heads scene of the 7-scenes dataset in 2-D. The anchor reference points are to be used by regression techniques to interpolate/extrapolate descriptors at target poses. Since in the case of extrapolation we do not subsample along the reference trajectory as in interpolation, there are nonanchor reference points in the extrapolation experiment but not in the interpolation experiment.

p_{new} can be selected on the path between the anchor poses p_{a1} and p_{a2} .

In the *extrapolation scheme*, the set of target extrapolation poses P' is selected in the vicinity of the poses in P , but not necessarily on a path between them. One possibility is to generate these target poses in a uniform grid within a certain distance threshold around each anchor. Another possibility is to define a single global uniform grid and only evaluate grid points using the nearest anchor points (within some distance threshold) similar to the work in [9]. The former approach leads to a denser grid, although it is globally nonuniform.

Examples of the reference, query, and target poses are shown in Fig. 2 to illustrate interpolation and extrapolation for map densification on the 7-scenes dataset [51].

C. Descriptor Regression Strategies

We consider several strategies to predict a new descriptor $f_{\text{new}} \in R'$ for a given target pose $p_{\text{new}} \in P'$ and the sparse

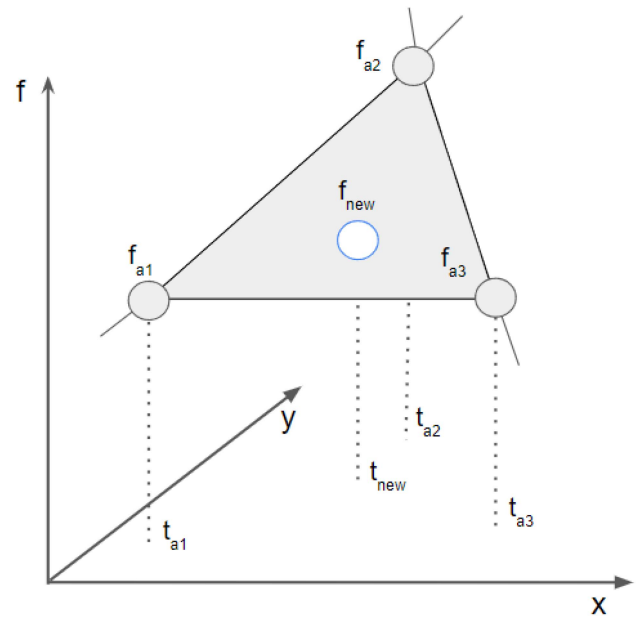


Fig. 3. Locally-fit plane given three anchor points in a 2-D world. Note that this plane is for a single feature dimension; so, in practice, there will be N such planes.

reference map M_{sparse} , which could be applied to the extrapolation and/or interpolation tasks. In principle, a regression method fits a model to express the dependent variable(s) as a function of the independent variables, thereby capturing the local trend in the space around the fitted samples. For feature descriptor regression, our objective is to express the feature space as a function of the pose. Since this feature space is latent, it is unclear to what extent we can assume it to be globally or locally linear for changing pose; hence, we consider both linear and nonlinear regression techniques for CoPR, as follows.

1) *Linear Interpolation*: The simplest strategy only applies to interpolation, where we only use the translation and not the orientation of each pose. We aim to predict the descriptor for an intermediate translation between two known translations. The target descriptor in this case is a linear weighted combination of its two anchors

$$f_{\text{new}} = (1 - \alpha_{a1}) \times f_{a1} + (1 - \alpha_{a2}) \times f_{a2} \quad (1)$$

$$\alpha_{a1} = \beta_1 / (\beta_1 + \beta_2) \quad (2)$$

$$\alpha_{a2} = \beta_2 / (\beta_1 + \beta_2) \quad (3)$$

where $\beta_1 = \|t_{\text{new}} - t_{a1}\|_2$, $\beta_2 = \|t_{\text{new}} - t_{a2}\|_2$, and f_{a1} , f_{a2} are the two anchor feature descriptors.

2) *Linear Regression Using Local Plane Fit*: As a second approach, we investigate a local plane fit to consider more anchors and allow extrapolation too. This also only uses the translation and not the complete pose. Given the target translation t_{new} , the O NN anchor points from M_{sparse} in terms of Euclidean translation distance are selected. For each descriptor dimension, a linear plane is least-squares fitted on the anchor values, and the plane is evaluated at the translations of the target t_{new} to regress f_{new} . This linear regression is abstractly depicted in Fig. 3 for a single feature dimension (f) in a 2-D pose space (x and y). Note

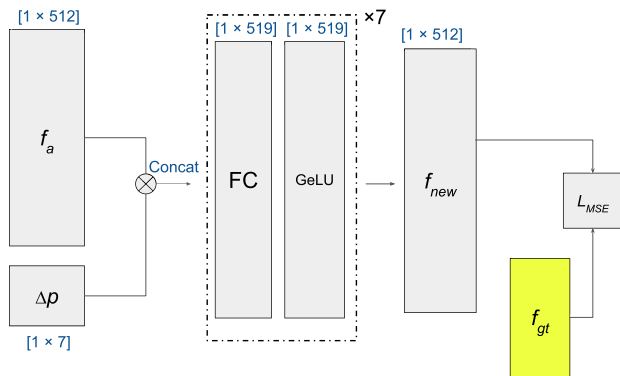


Fig. 4. Nonlinear-deep-learning-based model H that we train to regress the descriptor f_{new} at a target location. The input is an anchor reference descriptor f_a and the relative pose Δp between the anchor location p_a and the target location p_{new} .

that a more complex polynomial or spline regression could be used too, but we limit our approach to linear regression here as the most canonical implementation of this general approach.

3) *Nonlinear Regression Network*: In this strategy, we directly regress $f_{new} = H(f_a, \Delta p)$ from a single anchor descriptor f_a , and the relative pose Δp specifying the translation difference and the quaternion rotation between the anchor pose p_a and the target pose p_{new} . As nonlinear descriptor regressor H , we use a fully-connected deep neural network consisting of seven hidden layers with a GeLU [52] activation. The input to the network is the N -dimensional anchor feature descriptor f_a and the relative pose Δp stacked together while the output is the N -dimensional target feature descriptor f_{new} at the pose p_{new} . The dimensionality of the input layer and hidden layers is the same, i.e., $N + 7$, as the relative pose vector Δp has a length of 7 while the output layer has only N dimensions. This network is shown in Fig. 4. In preliminary experiments on Microsoft 7-scenes (see Section IV-A), we explored other activations and using fewer or more layers. We found GeLU works best and that the network can overfit with more than seven layers.

Given a pretrained and fixed encoder G for computing feature descriptors, the nonlinear regression network is trained on available descriptor pairs (e.g., an anchor descriptor f_a and a GT target descriptor f_{gt}) with known relative pose Δp between them, and a mean-squared error loss

$$L_{MSE} = \|H(f_a, \Delta p) - f_{gt}\|_2. \quad (4)$$

D. Losses for the Feature Encoder

Next, we discuss the choice for the training loss of the feature encoder G , since the feature space is key for the general localization quality and also defines the complexity of the regression task that map densification should solve. The feature encoder G takes as input an image I and computes its N -dimensional feature descriptor f_I . We will compare three different training strategies, namely training with a triplet loss [3], an RPE loss [5], and a distance-based loss [16], which are shortly summarized here.

For training with a *triplet loss*, the network computes N -dimensional feature descriptors $\{f_q, f_p, f_n\}$ for three images $\{I_q, I_p, I_n\}$: a query I_q , a positive match I_p with varied viewpoint, and a negative match I_n that represent a different scene/place. Each of these three N -dimensional feature descriptors is then normalized and penalized with a triplet loss. The triplet loss is the same as that of in [3], which penalizes the network given a Euclidean distance function $d_f(f_1, f_2) = \|f_1 - f_2\|_2$ and a margin m with a triplet loss

$$L_{\text{triplet}} = \max\{d_f(f_q, f_p) - d_f(f_q, f_n) + m, 0\}. \quad (5)$$

For the *RPE loss* [5], f_q and f_p are stacked together and passed through a relative-pose regressor consisting of fully-connected layers to output the estimated 6-DoF relative pose Δp_{est} between the two input images. The network is trained with a mean-squared error loss, i.e.

$$L_{\text{relative}} = \|\Delta p_{\text{est}} - \Delta p_{\text{gt}}\|_2 \quad (6)$$

given the GT relative pose Δp_{gt} . This is the same network as that of Laskar et al. [5]. To regress the relative pose Δp_{est} correctly, the network has to encode viewpoint information in the feature descriptors $\{f_q, f_p\}$. Nevertheless, this relative pose-based loss does not explicitly force the network to encode representations that encourage the closest descriptor in 3-D physical space to be the closest in feature space.

Therefore, the third loss is the *distance-based loss* L_{distance} as introduced in the work of Thoma et al. [16]

$$L_{\text{distance}} = \|\Delta f - \Delta t\|_2. \quad (7)$$

This loss explicitly penalizes the network based on the Euclidean distance Δf between feature descriptors $\{f_q, f_p\}$ and the Euclidean distance between their corresponding GT translation poses Δt .

E. Relating CoPR to RPE

Our main focus is the task of VPR for VL. Nevertheless, map densification can also improve the accuracy of CtF, i.e., VPR plus RPE [5]. This section expands on the methodological relation between CoPR and RPE.

Formally, given two feature descriptors f_1 and f_2 and the relative pose between their corresponding locations Δp , a CoPR strategy as in Section III-C3 models a function $f_2 = H(f_1, \Delta p)$. In contrast, RPE aims to learn a function $\Delta p = L(f_1, f_2)$. While these two functions H and L appear similar, these approaches have different benefits. A useful property of CoPR is that it can be done offline; thus, localization reduces to a single-stage image-retrieval problem at runtime while RPE is performed online and thus leads to a multistage CtF formulation.

A more crucial difference is that RPE assumes its two input images represent the same scene and, thus, must rely on the accuracy of the preceding image-retrieval step. Consider a query I_q taken in a scene A , e.g., a room in an office, and a sparse reference map containing various visually similar scenes, e.g., other rooms in the same office (see Fig. 5). The image-retrieval system might fail and retrieve a reference f_B from an arbitrarily distant scene (“room”) B instead of any nearby reference f_A from the

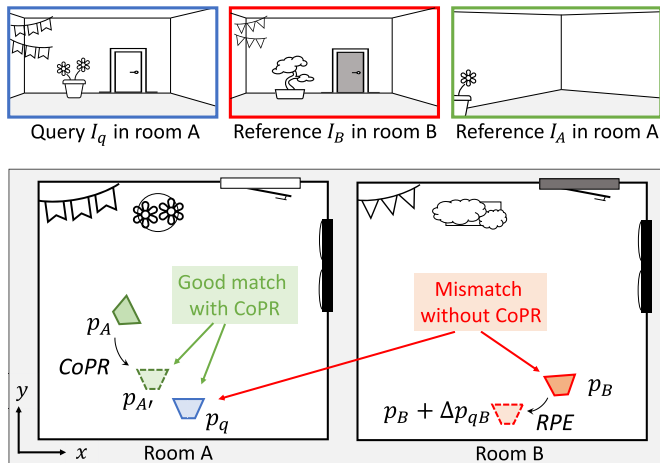


Fig. 5. Perceptual aliasing of rooms A and B: Query I_q in room A appears more similar to reference I_B in room B than to reference I_A in correct room A. If VPR retrieves the wrong reference f_B for f_q , RPE between f_B and f_q cannot correct this: The “apparent” difference between the query pose p_q and reference pose p_B is nearly zero. CoPR, therefore, aims to improve VPR instead by adding references for more diverse poses to the map, e.g., $f_{A'}$ for $p_{A'}$.

actual scene A, i.e., when $\|f_B - f_q\|_2 < \|f_A - f_q\|_2$. We refer to the inability to distinguish such similar scenes as *perceptual aliasing* [53]. These scenes should ideally all be represented as nearby references in the feature space, but in a sparse reference map, some scenes could be underrepresented, and retrieving the best (or even top- k) matches for a query might never include the correct scene. RPE cannot correct such retrieval failures. For instance, a pose difference between correct reference I_A and query I_q (both at room A) could limit the visual overlap between their images, making their descriptors f_A and f_q dissimilar. If the visual content of I_b and I_q appear more similar, their pose difference would *appear* relatively small, even though these are at completely different scenes. Since $\Delta p_{qB} = L(f_q, f_B)$ will just estimate the small *apparent* pose offset, RPE results in an incorrect final pose estimate for the query, $p_B + \Delta p_{qB}$.

By densifying the reference map, we can instead extend the references in room A to represent more diverse poses. A regressed descriptor $f_{A'}$ at a new pose $p_{A'}$ closer to the query than the original reference p_A can improve the best match, $\|f_{A'} - f_q\|_2 < \|f_B - f_q\|_2$, resulting in a good VPR localization estimate $p_q \approx p_{A'}$. We demonstrate the existence of this effect using constructed failure cases in our experiments of Section IV-G. In CtF localization, RPE afterward still reduces this gap further by estimating $\Delta p_{qA'} = L(f_q, f_{A'})$, such that $p_q = p_{A'} + \Delta p_{qA'}$. CoPR and RPE are, therefore, complementary techniques.

IV. EXPERIMENTS

In this section, we present our experimental setup in detail, including the datasets, baselines, and evaluation metrics. First, we validate using the encoder G_{distance} as our primary encoder. We then present our results of using descriptor regression for interpolation and extrapolation experiments. We show how different feature encoders can benefit from CoPR and the effect

of map density on localization performance. We also show the relation between CoPR and CtF localization and, finally, provide the computational details of our work.

A. Experimental Setup

Here, we explain the datasets, evaluation metrics, and the various parametric choices used in our experiments.

1) *Datasets*: We use three datasets for evaluation, Microsoft 7-scenes, the Synthetic Shop Facade, and the Station Escalator dataset. Our choice of these datasets is based on their wide adoption for evaluating VL in the existing literature as reviewed previously and their complementary nature: indoor versus outdoor, different levels of spatial coverage, and different types (parallel versus intersecting) of traversals. We discuss each dataset in turn.

Microsoft 7-scenes dataset [51] has been a long-standing public benchmark for 6-DoF indoor localization [5], [9], [54]. This dataset consists of seven different indoor scenes collected using a Kinect RGB-D camera and provides accurate 6-DoF GT poses computed using a Kinect Fusion [55] baseline. Each scene spans an area of a few square meters and contains multiple sequences/traverses (viewpoint-varied) within a scene. Each sequence itself then contains between 500 and 1000 images, where each image has a 640×480 pixels resolution. There are separate query and reference sequences, which contain novel viewpoints of the same scene. The images and poses in the query trajectory act as our training set for training both the feature encoder G and the nonlinear descriptor regressor H . The reference trajectory is further divided into two splits: 1) validation and 2) test sets, with 40% images in the validation set and 60% images in the test set. The validation set is used for validating the encoder G and the nonlinear regression network H at training time. This reference trajectory is then used for the interpolation and extrapolation experiments.

The *Synthetic Shop Facade* dataset proposed in [9] represents images and poses regressed from a 3-D model of a real-world outdoor shopping street [4] and consists of multiple sequences/traverses of a single scene. It contains about 9500 images at novel viewpoints with an image resolution of 455×256 pixels. There are separate splits for query and reference sequences that contain different viewpoints. The training, validation, and test sets follow the same strategy as that of the 7-scenes dataset.

The *Station Escalator* dataset proposed in [9] contains two parallel trajectories through a station and is, hence, useful for studying extrapolation benefits across parallel lanes. The dataset contains 330 query images and 330 reference images with an image resolution of 1557×642 pixels and 6-DoF accurate poses. For this dataset, we intend to regress descriptors from one trajectory (say A) to its parallel trajectory (say B), thus the nonlinear regression network H needs to be trained with such relative pose change between A and B. Therefore, given the two original parallel trajectories, we divide both into three parts: 1) training, 2) validation, and 3) test sets. The training images are selected as every 50th image in both trajectories while the remaining images are equally divided between the validation

and test sets. The training images from both traverses are used to train the descriptor regression models. For experiments, the validation and test images from trajectory A combined together act as our query images. The validation and test images from trajectory B in addition to the training images from trajectory A act as the reference images.

2) *Evaluation Metrics*: The evaluation metric is the median translation error (MTE) in meters and the median rotation error (MRE) in degrees over all the estimated query images' poses, as commonly used in existing literature [5], [9], [54]. The median is normally preferred over the mean since outliers can skew the latter by any amount. The translation error is the Euclidean distance between the query image's translation and the best-matched reference image's translation. The rotation error is the angular difference between the quaternion vectors of a query image and its best-matched reference image, as used in the reviewed literature.

3) *Training Details and Parametric Choices*: We use the output of the final global average pooling layer of a ResNet34 [56] backbone feature encoder, and thus a feature descriptor size of $N = 512$ is used throughout this work. The feature encoder G and the nonlinear descriptor regressor H are trained separately. For training all the three feature encoders G_{triplet} , G_{relative} , and G_{distance} and for nonlinear regression network H , we use the Adam optimizer for model optimization with learning rates of $1e^{-5}$, $1e^{-4}$, $5e^{-5}$, and $5e^{-4}$ for G_{triplet} , G_{relative} , G_{distance} , and H , respectively. The weights of the ResNet34 backbone are initialized via pretraining on ImageNet-1 K and fine-tuned on the datasets used in this work while the nonlinear regression network H is trained from scratch for each dataset.

For training the encoder G_{triplet} , images from the training sets of different scenes of the 7-scenes dataset are chosen randomly to act as negatives while images from the same scene with varied viewpoints are chosen as positives. We use a margin of $m = 0.3$ for the triplet loss, same as [3]. The feature encoder G is trained jointly on the training pairs of all the seven scenes in the 7-scenes dataset. The encoders trained using triplet loss (G_{triplet}) and RPE loss (G_{relative}) are only trained on the 7-scenes dataset and used for experiments on all the datasets while the model trained using distance-based loss (G_{distance}) is trained separately for each dataset. We later show the reasons behind this separate training for distance-based loss in Section IV-B.

A dedicated nonlinear regression model H is trained for each of the three datasets. The nonlinear regression model H trained for one dataset is used for both the interpolation and extrapolation experiments of that dataset. For the least-squares plane fit to linearly regress each feature dimension, $O=4$ is chosen as the number of NN anchors, which is the minimum number needed to fit a plane in 4-D (i.e., 3-D world plus 1-D feature).

B. Encoder Loss Function and Localization Accuracy

Here, we intend to understand the first part of the two potential requirements for accurate VPR-based localization: viewpoint variance. The encoder training objectives favoring viewpoint variance can have a considerable effect on the VPR-based

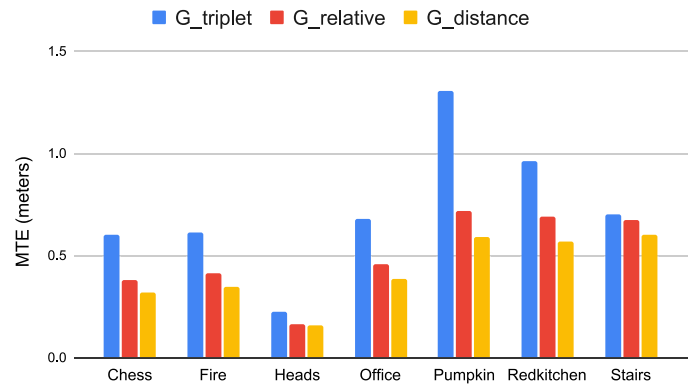


Fig. 6. MTE of the three encoders when used for performing VPR-based localization on all the scenes of the 7-scenes dataset. Training with distance-based loss leads to lower MTE than other losses.



Fig. 7. MTE of the three encoders used for testing VPR-based localization on the Synthetic Shop Facade dataset, when trained on the same and different dataset. Notably, G_{triplet} and G_{relative} trained on the 7-scenes can outperform G_{distance} trained on the 7-scenes dataset. However, G_{distance} when trained and tested on the Synthetic Shop Facade dataset performs the best. Since the Shop Facade dataset contains images of only one scene, unlike the 7-scenes dataset, we could not select proper negative images in this dataset and do not train G_{triplet} on this dataset.

localization error. The change in localization error for G_{triplet} , G_{relative} , and G_{distance} is shown in Fig. 6 for the 7-scenes dataset, where a distance-based loss leads to the lowest localization error. This localization error is without map densification and is purely the effect of different training objectives for the encoder G .

Moreover in Fig. 7, we observe the (de)generalization of these feature encoders from one dataset to the other. This is done by evaluating the VPR-based localization performance of a given encoder on datasets other than the training dataset for a given model. We note that the network G_{distance} trained on the 7-scenes dataset does not perform well on the Shop Facade dataset and is outperformed by G_{triplet} and G_{relative} trained on the 7-scenes dataset, which suggests that G_{distance} is less generalizable. We, therefore, train G_{distance} on the Shop Facade dataset, after which it outperforms the other networks. This degeneralization of distance-based loss has also been reported in [16] and an intuitive explanation could be that distance-based

losses are more sensitive to structural changes between different domains and the change in scene appearance with changing scene depth.

Since distance-based loss leads to the lowest localization error, we only use G_{distance} as our backbone encoder for the experiments in Sections IV-C and IV-D. However, we later show in Section IV-E that all the encoders (G_{triplet} , G_{relative} , and G_{distance}) can benefit from CoPR, albeit at varying levels of accuracy.

C. Extrapolation Experiments

We first explain the setup used for extrapolation experiments, followed by the extrapolation methods and baselines, and then the corresponding results and discussion.

1) *Extrapolation Setup*: We use all three datasets to examine the effects of extrapolation. All of these three datasets have properties useful for our CoPR analysis. Thus, we first explain the setup for extrapolation on these three datasets, as follows.

The extrapolation experiments are performed on all scenes of the *7-scenes dataset*. For each scene in the *7-scenes dataset*, there are multiple reference sequences; thus, we take one of the reference traverses/sequences as our anchor reference trajectory. We then discard the remaining reference sequences³ to get the original sparse map M_{sparse} . Then, on the selected reference sequence, we select every K th sample (where $K = 50$) as our anchor point. Then, for each anchor point, we sample target points uniformly in the x and y directions keeping the viewing direction and z fixed to get the dense extrapolated map M_{dense} . The sampling of target points is done with a fixed step size e_{step} and a maximum spatial span e_{span} for extrapolation. We use a step size of $e_{\text{step}} = 0.05$ meters for all seven scenes and the spatial span e_{span} is set to cover the complete area of the scene. Examples of this extrapolation are shown in Fig. 2 for the *7-scenes dataset*.

The *Synthetic Shop Facade dataset* provides a query sequence, a single anchor reference sequence, and multiple target reference points sampled uniformly over a fixed grid across this anchor reference sequence. We use this already provided distinction to get M_{sparse} and M_{dense} . The query, anchor, and target extrapolated points contain novel viewpoints of the same scene and we refer the reader to the figure [9] here⁴ for visualization of the scene and target point distribution.

In the case of the *Station Escalator dataset*, the anchor reference images act as the sparse reference map M_{sparse} . Extrapolation on the *Station Escalator dataset* is straightforward: All images on the reference trajectory act as our anchor points and we regress a target descriptor using each anchor at an offset of 1.8 m on the x -axis from the anchor reference pose. Then, the target descriptors combined with M_{sparse} descriptors act as our extrapolated map M_{dense} .

2) *Extrapolation Methods*: Two descriptor regression methods are compared for extrapolation. *Linear Regression (Lin. Reg.)* is the local plane fit method introduced in Section III-C2. For the *7-scenes* and the *Shop Facade dataset*, the O NN anchor

points are selected from the reference trajectory, and for the *Station Escalator dataset*, we select two NN anchor points from each of the two parallel trajectories A and B.

Nonlinear Regression Network (Non-lin. Reg.) is the neural network regression approach from Section III-C3.

Extrapolation Baselines. Sparse Map: The primary baseline for extrapolation is the sparse map M_{sparse} , where feature descriptors are only available at sparse poses P .

3-D model: As mentioned in Section IV-A, the *Shop Facade dataset* already provides distinct anchor reference points and target extrapolation points. Since the images for these target extrapolation points are already available, their corresponding feature descriptors at all poses in the extrapolated map can also be computed. We refer to this method as *3-D Model* in our results, where the feature descriptors at all locations (anchor and nonanchor) in M_{dense} are computed using G_{distance} and no descriptor is regressed. This baseline [9] helps us to understand how well our extrapolation performs in comparison to having the GT images at all locations in the extrapolated map. *Oracle retrieval*: We also show the minimum possible translation error and the corresponding rotation error obtained by an oracle retrieval method, which always retrieves the GT 3-D Euclidean closest match in the extrapolated map M_{dense} . These errors indicate the VPR base errors for the used queries, and would only be zero if the query poses coincide with the reference poses in the map.

4) *Extrapolation Results*: We report the extrapolation results in Table I for the originally sparse, linearly extrapolated, and nonlinearly extrapolated maps for all the seven scenes in the *7-scenes dataset*. The matches between the query and the reference trajectories for the extrapolation experiment are shown in Fig. 8 for the *Stairs scene* of the *7-scenes dataset* as an example. It can be seen that extrapolation leads to significant performance improvement over no extrapolation in terms of a translation error. By using extrapolation we match descriptors closer to the query trajectory. We also note that the nonlinear regression model H performs better than the linear regression model, indicating that extrapolating across the trajectory requires a nonlinear approach to handle the complexity of the feature space. We do not see performance improvement in translation error due to extrapolation on the *Heads scene*, where the query and the reference trajectories are already relatively close to each other compared to the other scenes. Moreover, we observe that with the current map densification setup, we cannot improve angular estimation. However, it is important to notice that even retrieving the Euclidean closest match in physical space leads to an increase in rotation error, as shown by *Oracle retrieval* in Tables I and II. We further discuss this increase in rotation error and the reasons behind it in Section V.

The same findings are extended to the *Synthetic Shop Facade dataset* as reported in Table II. We see performance improvement thanks to extrapolation and the nonlinear regression model H outperforms linear regression. We also observe that the VPR performance of the nonlinearly extrapolated map (*Non-lin. Reg.*) is similar to the map densified using 3-D modeling, which suggests that the trained nonlinear regression model H closely regresses the original descriptors, without access to the images at the target poses.

³If we do not discard other reference sequences during the extrapolation experiment, they overlap with target extrapolated/regressed descriptors and make the experimental setup less challenging.

⁴[Online]. Available: https://github.com/tsattler/understanding_apr

TABLE I
EXTRAPOLATION EXPERIMENTS ON THE 7-SCENES DATASET

Metric	Map	Densification	Retrieval	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Avg.
MTE (m)	M_{dense}	-	<i>Oracle</i>	0.083	0.070	0.030	0.072	0.077	0.216	0.119	0.095
MTE (m)	M_{sparse}	-	VPR	0.318	0.348	0.158	0.383	0.589	0.567	0.600	0.423
MTE (m)	M_{dense}	<i>Lin. Reg.</i>	VPR	0.245	0.310	0.163	0.338	0.426	0.444	0.532	0.351
MTE (m)	M_{dense}	<i>Non-lin. Reg.</i>	VPR	0.167	0.279	0.159	0.264	0.346	0.427	0.430	0.296
MRE (°)	M_{dense}	-	<i>Oracle</i>	28.44	25.56	21.25	58.37	56.33	35.97	23.85	35.68
MRE (°)	M_{sparse}	-	VPR	22.54	20.88	16.49	38.89	44.89	34.65	24.32	28.95
MRE (°)	M_{dense}	<i>Lin. Reg.</i>	VPR	29.04	18.49	16.62	39.10	61.96	33.00	25.41	31.95
MRE (°)	M_{dense}	<i>Non-lin. Reg.</i>	VPR	26.87	22.02	16.54	47.95	58.90	36.33	21.29	32.84

The MTE and MRE are reported. The oracle retrieval shows the minimum achievable MTE and the corresponding MRE. Best in bold.

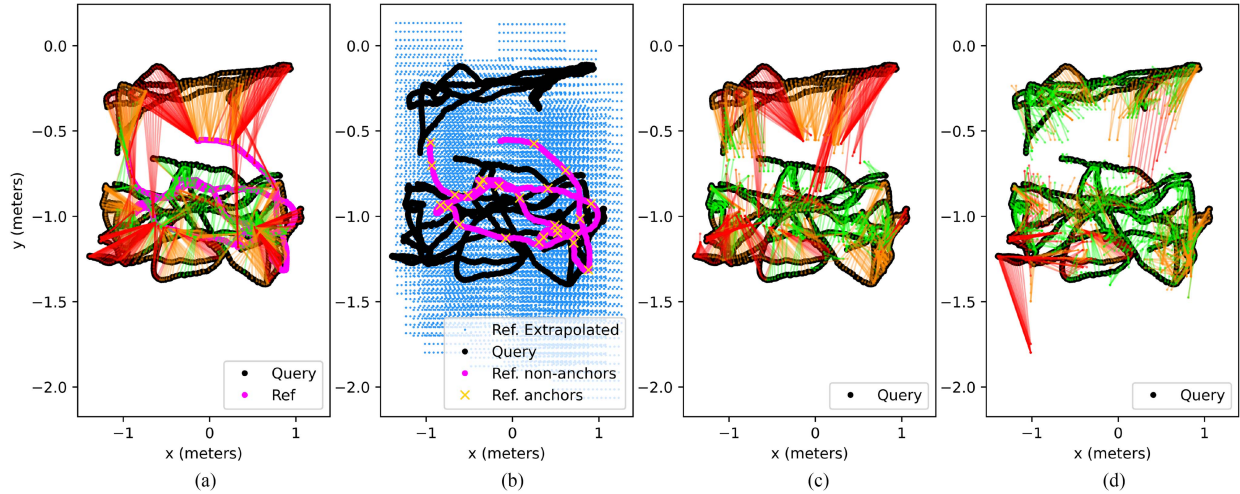


Fig. 8. Extrapolation experiments on the Office scene of the 7-scenes dataset. (a) Matches between the query and the reference points for the sparse map M_{sparse} . (b) Poses in the densified map M_{dense} . (c) Matches in map densified using *Lin. Reg.* (d) Matches in map densified using *Non-lin. Reg.* All matches are color-coded as *green*, *orange*, and *red* with the increasing 3-D Euclidean distance in the physical space. The reference poses in (c) and (d) are the same as in (b) and, thus, are not shown to avoid cluttering. The nonlinearly densified map (d) clearly leads to better performance than other maps, albeit with some failure cases toward the bottom-left of the plot.

TABLE II
EXTRAPOLATION EXPERIMENTS ON THE SYNTHETIC SHOP FACADE AND THE STATION ESCALATOR DATASETS

Metric	Map	Densification	Retrieval	Shop Facade	Station Escalator
MTE (m)	M_{dense}	-	<i>Oracle</i>	0.188	0.26
MTE (m)	M_{sparse}	-	VPR	0.705	2.17
MTE (m)	M_{dense}	<i>3-D Model</i> [9]	VPR	0.335	NA
MTE (m)	M_{dense}	<i>Lin. Reg.</i>	VPR	0.541	2.10
MTE (m)	M_{dense}	<i>Non-lin. Reg.</i>	VPR	0.344	0.94
MTE (°)	M_{dense}	-	<i>Oracle</i>	11.25	9.45
MTE (°)	M_{sparse}	-	VPR	10.99	8.54
MTE (°)	M_{dense}	<i>3-D Model</i> [9]	VPR	11.13	NA
MTE (°)	M_{dense}	<i>Lin. Reg.</i>	VPR	10.99	8.60
MTE (°)	M_{dense}	<i>Non-lin. Reg.</i>	VPR	11.13	8.99

The MTE and MRE are reported. The oracle retrieval shows the minimum achievable MTE and the corresponding MRE. Best in bold.

The results on the Station Escalator dataset also support the motivation of this work, since we are able to significantly improve the localization accuracy, as reported in Table II. We also show the qualitative results on the Station Escalator dataset in Fig. 9. These results highlight the utility of descriptor regression in cases where parallel traverses are common, such as highway lanes, train tracks, escalators, and many such laterally viewpoint-varied paths.

We observe more benefits of nonlinear descriptor regression on the Station Escalator dataset than on other datasets. Linear

regression does not work well on this dataset; the selected anchor poses are too distant from the query trajectory. Recall that for this dataset the training pairs include sparse samples (every K th image) from both the query and reference traverses to increase the variance in the training data, as there are only two traverses in total in this dataset. Still, our extrapolation experiments do not extrapolate to the exact query locations but to close-by locations. We observe that training with similar relative pose differences as those observed at test time leads to performance benefits. In a real-world application, if only sparsely sampled

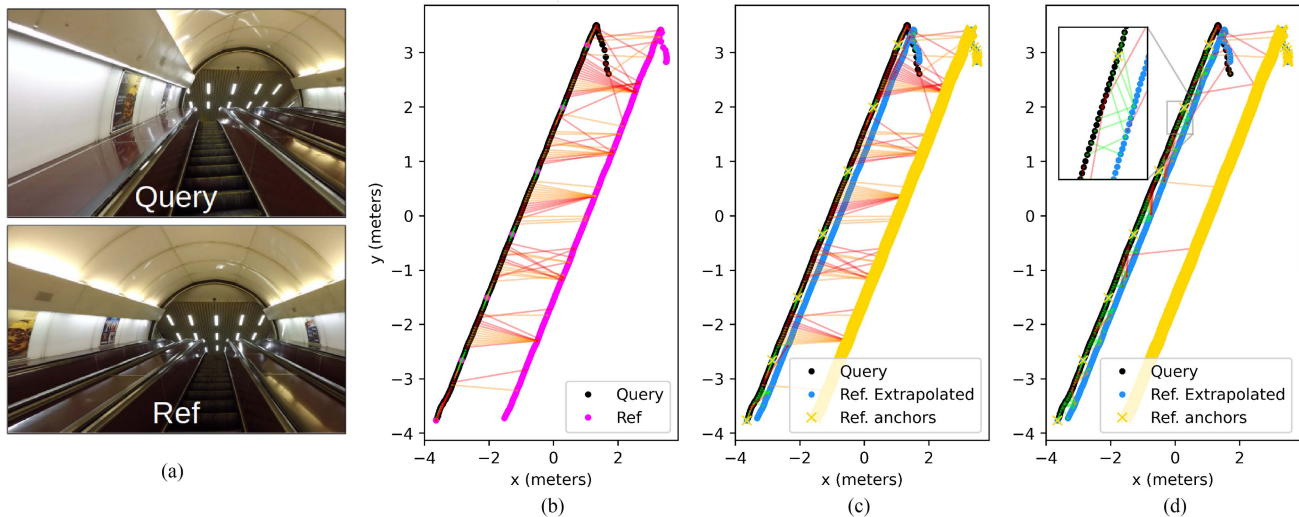


Fig. 9. Extrapolation experiments on the Station Escalator dataset. (a) Exemplar query and reference images. Then, the matches between the query and the reference points for the (b) original sparse map M_{sparse} , (c) linearly regressed (*Lin. Reg.*) map M_{dense} , and (d) nonlinearly regressed (*Non-lin. Reg.*) map M_{dense} . These matches are color-coded as *green*, *orange*, and *red* with increasing 3-D Euclidean distance in the physical space. Extrapolation with nonlinear regression network H is done using only the points on the anchor reference trajectory in *yellow* on the right, whereas the sparse anchor points in *yellow* on the query trajectory are only used at training time.

images are collected for parallel trajectories, the pose differences are representative to train a regression model and densify the trajectories for improved localization accuracy.

D. Interpolation Experiments

We now explain the setup used for interpolation experiments, followed by the methods and baselines and, then, the corresponding results and discussion.

1) *Interpolation Setup*: We perform the interpolation experiments on all the scenes in the 7-scenes dataset. Similar to the extrapolation setup, interpolation uses the same concept of a sparse map M_{sparse} and a dense map M_{dense} ; although for the interpolation experiments, these maps are defined differently than for the extrapolation experiments. For interpolation, the full reference trajectory of a scene is used as the GT dense map M_{dense} . We then subsample the reference trajectories by a factor of $K = 50$, such that the consecutive images in a trajectory still contain visual content overlap. This reduced set of references is used as the sparse map M_{sparse} . The GT dense map serves as a baseline that can assess the performance of VPR if densely sampled reference images would be available while the subsampled version shows the performance when only a sparse set of reference images are available. Examples of this subsampling are shown in Fig. 2. For CoPR, the poses in P from the sparse map act as our anchor poses while the additional poses P' found in the GT dense map act as the target poses. All feature descriptors in M_{sparse} and the query descriptors are computed using the feature encoder G_{distance} explained in Section III-D.

2) *Interpolation Methods*: The compared descriptor regression methods are the simple *Linear Interpolation (Lin. Interp.)* from Section III-C1; the *Linear Regression (Lin. Reg.)* from Section III-C2; and the *Nonlinear Regression Network (Non-lin. Reg.)* from Section III-C3.

3) *Interpolation Baselines. Sparse map*: The primary baseline for interpolation is the sparse map M_{sparse} , where feature descriptors are only available at sparse poses P .

GT dense map: Unlike the extrapolation experiments where we do not have true images (and hence descriptors) available at target poses, in the case of interpolation experiments, we do have these true images. Thus, this GT dense map M_{dense} is a baseline that serves the true descriptors for the target poses.

Oracle retrieval: We also show again the minimum possible translation error and the corresponding rotation error from the oracle retrieval method, as defined in Section IV-C3.

4) *Interpolation Results*: The results for all the methods and baselines for the interpolation experiment on the 7-scenes dataset are reported in Table III for all the seven scenes. The VPR matches between the query and reference trajectories for the Heads scene are shown in Fig. 10. We can see a general decrease in localization error when moving from the sparse map M_{sparse} to the GT dense map M_{dense} . Interestingly, we also see that even simple linear regression (*Lin. Reg.* and *Lin. Interp.*) can solve this problem well and is often the best-performing technique. Note though that linear regression is done using multiple anchor points which constraints the problem setting while the nonlinear regression network H only uses one anchor point. Nevertheless, this experiment shows that map densification even via interpolating along the trajectory is helpful, although has lesser benefits than extrapolation across the trajectory.

We will discuss the observed differences between the interpolation and extrapolation experiments in more detail in Section V.

E. Map Densification With Different Feature Encoders

Next, we test that using the nonlinear regression model H for extrapolating across anchor points is beneficial for all discussed feature encoders. This is reported in Table IV. However, the

TABLE III
INTERPOLATION EXPERIMENTS FOR THE 7-SCENES DATASET AT $K = 50$

Metric	Map	Densification	Retrieval	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Avg.
MTE (m)	M_{dense}	-	<i>Oracle</i>	0.109	0.183	0.097	0.117	0.115	0.129	0.132	0.126
MTE (m)	M_{dense}	<i>GT Map</i>	VPR	0.165	0.255	0.158	0.207	0.242	0.219	0.261	0.215
MTE (m)	M_{sparse}	-	VPR	0.210	0.322	0.212	0.237	0.250	0.271	0.263	0.252
MTE (m)	M_{dense}	<i>Lin. Interp.</i>	VPR	0.170	0.277	0.202	0.211	0.257	0.220	0.257	0.227
MTE (m)	M_{dense}	<i>Lin. Reg.</i>	VPR	0.169	0.257	0.165	0.216	0.214	0.224	0.262	0.215
MTE (m)	M_{dense}	<i>Non-lin. Reg.</i>	VPR	0.178	0.264	0.184	0.221	0.259	0.260	0.278	0.234
MRE ($^{\circ}$)	M_{dense}	-	<i>Oracle</i>	22.81	26.65	20.91	43.56	37.58	31.31	29.71	30.36
MRE ($^{\circ}$)	M_{dense}	<i>GT Map</i>	VPR	17.69	19.71	16.49	32.13	36.24	22.49	19.55	23.47
MRE ($^{\circ}$)	M_{sparse}	-	VPR	20.75	19.15	19.34	35.01	33.96	27.27	19.16	24.94
MRE ($^{\circ}$)	M_{dense}	<i>Lin. Interp.</i>	VPR	21.73	20.65	17.93	34.65	39.15	26.27	19.92	25.75
MRE ($^{\circ}$)	M_{dense}	<i>Lin. Reg.</i>	VPR	20.09	20.00	17.20	35.63	34.00	24.16	19.72	24.40
MRE ($^{\circ}$)	M_{dense}	<i>Non-lin. Reg.</i>	VPR	22.09	19.45	20.13	39.73	39.03	27.17	20.25	26.83

The MTE and MRE are reported. The oracle retrieval shows the minimum achievable MTE and the corresponding MRE. Best is in bold.

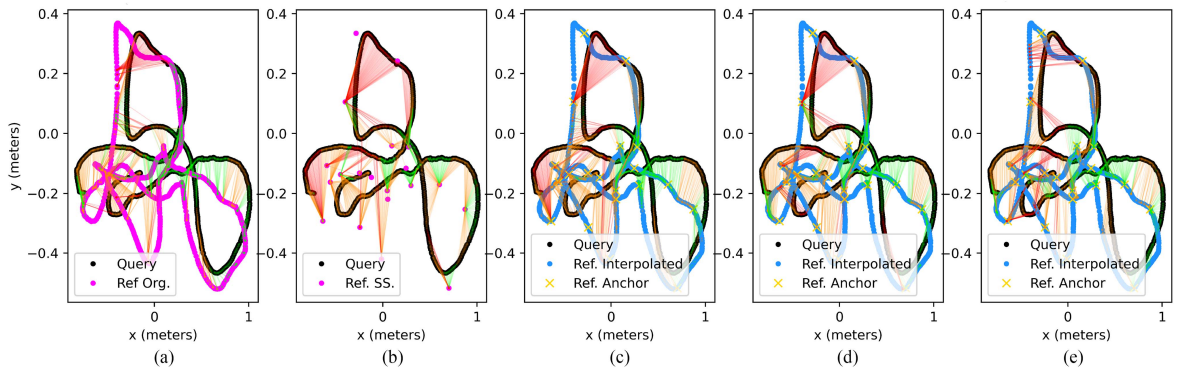


Fig. 10. Interpolation experiments on the Heads scene of the 7-scenes dataset. The matches between the query and the reference trajectories in (a) the GT dense map M_{dense} , (b) the sparse map M_{sparse} , (c) the linearly regressed (*Lin. Reg.*) map M_{dense} , (d) the linearly interpolated (*Lin. Interp.*) map M_{dense} , and (e) the nonlinearly regressed (*Non-lin. Reg.*) map M_{dense} given $K = 50$. The matches are color-coded as *green*, *orange*, and *red* with the increasing 3-D Euclidean distance in the physical space.

TABLE IV
EFFECT OF CoPR ON DIFFERENT FEATURE ENCODERS ON ALL SCENES FROM THE 7-SCENES DATASET, AND ON THE SYNTHETIC SHOP FACADE DATASET

Feature encoder	$G_{triplet}$		$G_{relative}$		$G_{distance}$	
	M_{sparse}	M_{dense}	M_{sparse}	M_{dense}	M_{sparse}	M_{dense}
7-scenes - Chess	0.600	0.450	0.379	0.260	0.318	0.167
7-scenes - Fire	0.612	0.542	0.414	0.296	0.348	0.279
7-scenes - Heads	0.227	0.215	0.166	0.147	0.158	0.159
7-scenes - Office	0.680	0.589	0.455	0.246	0.383	0.264
7-scenes - Pumpkin	1.306	1.208	0.720	0.479	0.589	0.346
7-scenes - Redkitchen	0.960	0.783	0.691	0.451	0.567	0.427
7-scenes - Stairs	0.699	0.780	0.673	0.374	0.600	0.430
Synthetic Shop Facade	2.234	1.419	2.219	1.641	0.705	0.344
Average	0.915	0.748	0.715	0.487	0.458	0.302

In the case of $G_{triplet}$, we use the triplet loss as motivated by Arandjelovic et al. [3] but do not use the VLAD descriptor module to keep the backbone the same for a fair comparison across all encoders.

corresponding localization accuracy is limited by the localization performance of the respective feature encoder. The MTE is reported for all three types of feature encoders on the sparse map M_{sparse} and the nonlinearly regressed (*Non-lin. Reg.*) map M_{dense} for the 7-scenes dataset and the Synthetic Shop Facade dataset. Such a generic boost of performance using map densification supports that CoPR can utilize inherent benefits of different types of feature encoders, for example, the domain generalization of $G_{triplet}$ and $G_{relative}$, and the viewpoint variance of $G_{distance}$.

F. Map-Density Versus Localization Accuracy

The motivation presented in this work suggests that the denser the reference map, the lesser will be the localization error of a

VPR-based localization system. In our work, this map density is modeled with the step size e_{step} . Therefore, in this section, we show the effect of increasing map density on the localization error by using extrapolation with nonlinear regression model H and feature encoder $G_{distance}$ for the 7-scenes dataset. This direct relation between the step size e_{step} and the MTE is presented in Fig. 11. Decreasing the step size leads to denser extrapolated maps, which then leads to a decrease in MTE for the nonlinearly extrapolated (*Non. Lin. Reg.*) map M_{dense} . The performance benefits for the scenes depend on the underlying scene geometry and the quality of descriptor regression. For example, in the case of Heads scene, the query poses and the sparse reference poses in M_{sparse} are already close to each other; thus, we do not see any performance benefits due to densification. While in other

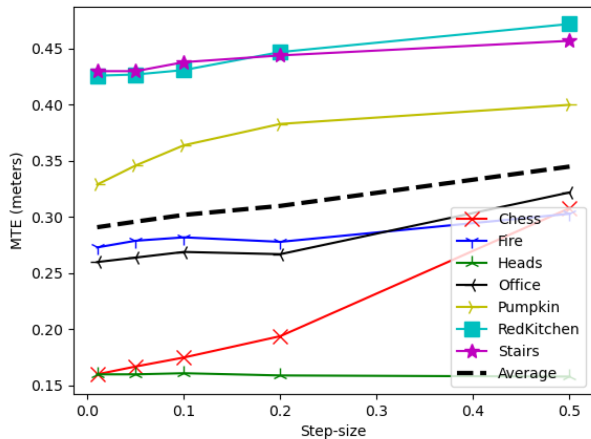


Fig. 11. Increase in MTE by increasing the step size e_{step} for all scenes of the 7-scenes dataset. A larger step size leads to sparser maps, which increases the translation error, whereas a smaller step size leads to denser maps, which are useful for accurate localization.

scenes, we see that map densification is helpful and is related to the level of map densification modeled with the step size e_{step} .

G. Benefits of CoPR for RPE

In this section, we look into the relation of CoPR with RPE and, hence, CtF localization, as discussed in Section III-E. In this experiment, we make this argument concrete by illustrating that situations exist where a sparse map leads to incorrect coarse retrieval of a visually similar image descriptor taken at an arbitrarily far location, which can in turn lead to the failure of CtF approaches. We argue that this error source is fundamentally due to the retrieval step, not due to the subsequent RPE step, and demonstrates that map densification could tackle this error source in some cases.

We create exemplar cases in the 7-scenes dataset where such an effect can be easily observed. A reference database of four sparsely sampled reference images around a query image is created for a given scene and a fifth *stray* reference image is added to this reference database. This stray image is taken from a completely different scene that has no real physical overlap with the query image. We use the feature encoder G_{relative} for image retrieval and the nonlinear descriptor regression network H to regress the expected descriptor at the query location given the nearest anchor reference descriptor. This regressed descriptor acts as the descriptor for a hypothetical 6th image in the reference database at the query location.

The objective of this experiment is to show that in the absence of the regressed descriptor, the stray image is selected as the best match for the query image, whereas in the presence of the regressed descriptor, the stray image is pushed downward in the list of retrieved images ranked by their matching scores. Note that in the case where the stray image is chosen as the best match, the localization error can be arbitrarily large, as a different scene can be quite far. We show four such example cases in Fig. 12 from the 7-scenes dataset, where we can observe that in the absence of the regressed descriptor, the stray image is chosen as the best

TABLE V
COMPUTATIONAL FOOTPRINT OF COPR, SEE ACCOMPANYING TEXT FOR DETAILS

	Map	7-scenes	Shop Fac.	Stat. Esc.
t_{train} (s)	-	510	540	960
t_{dense} (ms)	-	12.8	2,241	0.32
t_{enc} (ms)	-	6.16	8.39	5.88
t_{match} (ms)	M_{sparse}	0.02	0.1	0.02
	M_{dense}	0.05	0.32	0.08
t_{retr} (ms)	M_{sparse}	6.18	8.49	5.90
	M_{dense}	6.21	8.71	5.96
Map Size (#)	M_{sparse}	1000	231	337
	M_{dense}	13000	2531	667

match by the image-retrieval system. Since such stray cases are shown to exist in multiple scenes of the 7-scenes dataset, which is a small-scale dataset, this effect would amplify even further in spatially larger scenes due to the increased chances of perceptual aliasing.

Thus, without CoPR, sparse reference maps *could* lead to incorrect coarse retrieval, where the coarse pose estimate can be arbitrarily far-away and hence cannot be corrected by CtF approaches. By using CoPR, reference descriptors of the correct scene now appear close to the query descriptor. Finding all references near the query in the feature space thus identifies similar scenes, allowing to at least represent localization ambiguity and ideally obtain a correct best match. Without CoPR only the incorrect scene would have matched the query. Better retrieval also benefits CtF approaches, since the RPE step is only valid if the retrieved reference pose represents the correct scene. These constructed cases illustrate that CoPR and CtF are complementary approaches to improve VPR-based localization accuracy. Note that this analysis does not demonstrate that CoPR prevents false positives as a general rule, but that it is possible to construct cases where the complementarity of CoPR and CtF can be observed. Future works may investigate this further.

H. Computational Details

Finally, we report the sizes of the sparse and dense maps, the time spent t_{dense} on creating the dense maps M_{dense} using H , and the training times t_{train} of model H for all the datasets in Table V. For the 7-scenes dataset, the results are reported for the Office scene. The retrieval time t_{retr} in VPR is the sum of the time t_{enc} required to encode a query image into a feature descriptor and the time t_{match} spent to find the NN match of this descriptor in the map. Since the encoding time is several times higher than the efficient NN search, the retrieval time is not too affected by map densification. Note that the timings are not comparable between the datasets due to differences in map content (i.e., descriptors).

V. DISCUSSION

In this section, we identify the major limitations of our work and areas that need further investigation.

Angular error: In both the interpolation and extrapolation experiments, it is clear that our approach does not improve angular localization accuracy, as reported in Tables I, II, and III. However, it is also important to note that retrieving the

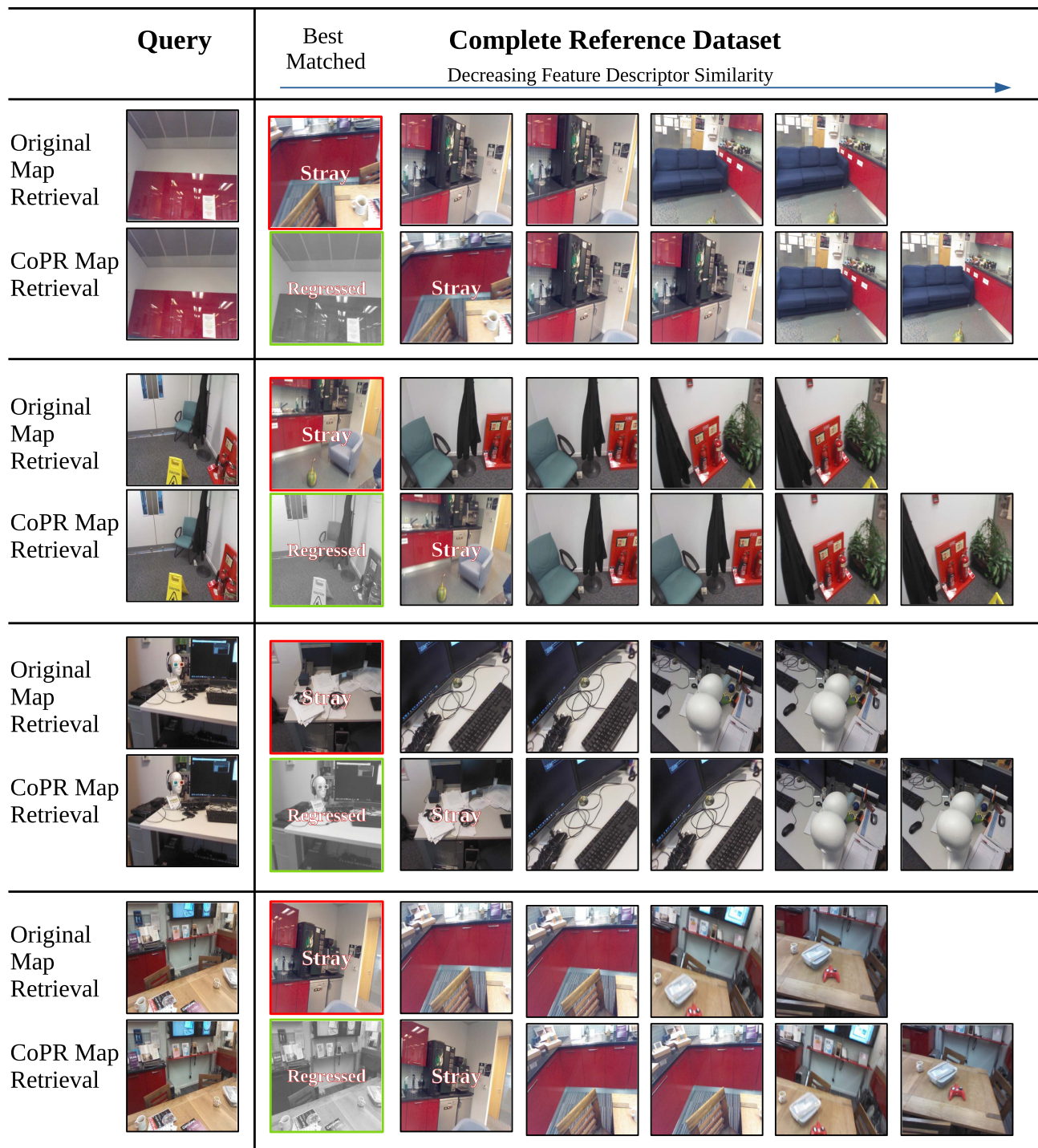


Fig. 12. Exemplar cases where image retrieval fails to retrieve useful coarse estimates for RPE in a sparse reference map. By regressing the expected descriptor at the query pose, we show that map densification could lead to robustness against such failure cases. The grayscale image in the reference set is only added for the reader's reference and represents only a hypothetical image for the regressed descriptor at the query pose since we do not synthesize images but only regress image descriptors. The *green* bounding box represents a correct match and the *red* bounding box represents an incorrect match.

GT Euclidean closest match in the physical space also leads to an *increase* in angular error (MRE). This is because the nearest match in terms of the translation may not have the same 3-D orientation. Thus, we attribute the increase in rotation error using CoPR to two reasons: First, during interpolation and

extrapolation experiments, we do not change the angular pose but only the translation pose, given the anchor points, for the target points, and second, the encoder G_{distance} does not optimize for angular localization error in its training objective. Thus, reducing both the translation and angular error requires that the

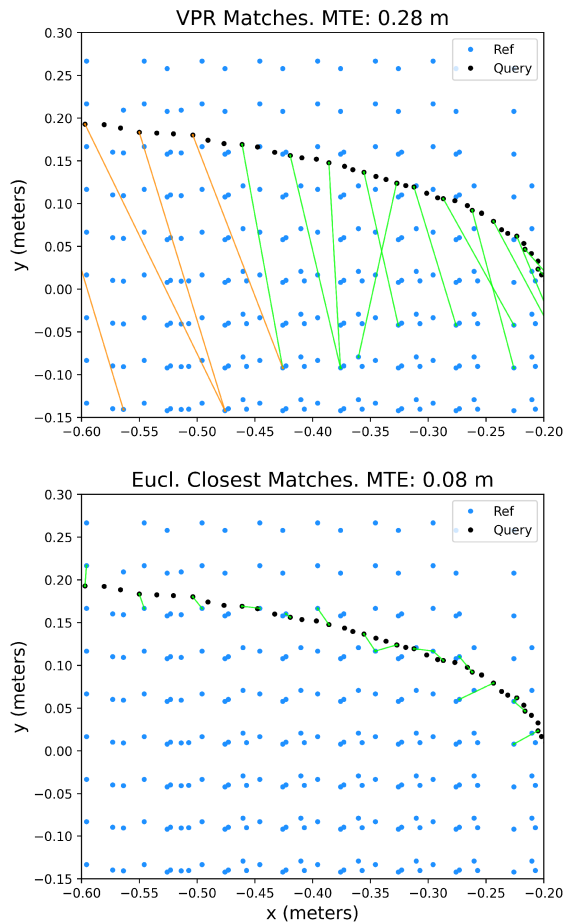


Fig. 13. VPR matches (top) and GT 3-D Euclidean closest matches in the physical space (bottom) between the query and the reference trajectories in the Fire scene of the 7-scenes dataset for the nonlinearly extrapolated (*Non-lin. Reg.*) map M_{dense} . The matches are color-coded as *orange* and *green* with increasing 3-D Euclidean distance. Although nonlinearly regressed target poses (in *blue*) are matched to by VPR, these are not always the Euclidean closest matches in the physical space. Hence, there is still room for improvement.

Euclidean closest match in the physical space has the closest angular orientation to the query image. Future works could look into the benefits of using distance+orientation-based encoder loss along with map densification in a 6-DoF setting.

GT closest matches: Our results on extrapolation show that map densification can lead to a significant decrease in localization error. Moreover, the extrapolation experiments on the Shop Facade dataset also show that the localization performance on the nonlinearly extrapolated (*Non-lin. Reg.*) map M_{dense} is close to the localization performance on a GT (obtained using 3-D modeling) dense map M_{dense} . However, the localization error given in the encoder G_{distance} and the nonlinear regression network H is still higher than the minimum possible localization error. We have reported the minimum possible translation error (*Oracle Retrieval*) in M_{dense} in Tables I, II, and III. We further show qualitatively in Fig. 13, the performance that could be achieved by an oracle VPR system that always retrieves the Euclidean closest match in the physical space as the best match in a dense map. This gap in performance presents room for future research in this area. Furthermore, our results only show

the generalization of nonlinear data-driven regression model H across viewpoints within the same scene; however, generalization across scenes could be the new frontier for CoPR.

Interpolation versus extrapolation: From our results of the two experiments, it can be noted that the absolute decrease in localization error from interpolation is less than the decrease in localization error from extrapolation. We hypothesize the following two reasons for this:

- 1) the query trajectory has a larger relative pose distance to the extrapolated poses than to the interpolated poses,
- 2) the viewpoint variance versus invariance of VPR encoders (as explained in Section III-D) acts as a bottleneck, since the VPR system does not necessarily match to the GT Euclidean closest match in the physical space but to *one of the closest* matches. We expect that major performance benefits, given these experiments, require models that have even better viewpoint variance than the feature encoder G_{distance} . This motivates viewpoint-variant VPR for high accuracy, in addition to the existing trends for viewpoint-invariant VPR [57].

Generally, we find that extrapolation is more useful than interpolation when a repeated traversal could occur at a laterally offset-ed path. Such trajectories are common to observe in real-world, for example, parallel traverses in outdoor scenes (Shop Facade dataset) and parallel traverses in indoor scenes (Station Escalator dataset). Other examples include lanes on a highway and parallel paths in corridors. However, our results do show that both interpolating and extrapolating descriptors generally give better localization accuracy than using sparser reference maps, which suggests that map densification (CoPR) along the trajectory and/or across the anchor points can be useful for VPR.

VI. CONCLUSION

In this article, we investigated the discrete treatment of places in a VPR map. We have shown that map densification whether using interpolation or extrapolation is helpful to reduce translation error. Our results for the 7-scenes dataset suggest that interpolating along the trajectory is an easier problem and can be solved with simple linear regression in the local neighborhood, however, extrapolation benefits from a nonlinear treatment. Moreover, our proposed nonlinear regression network only uses a single anchor point for regression while our linear regression method uses multiple anchor points. We validated that map densification is helpful for feature encoders trained with the three different types of losses and that the highest accuracy is achieved when using a distance-based loss. Moreover, the benefit of map densification is shown for three datasets: 1) 7-scenes, 2) Synthetic Shop Facade, and 3) Station Escalator, where each of them represents a different type of problem setting. We also discussed that RPE and CoPR address related but complementary problems. We demonstrated through several constructed cases that in a sparse map localization might fail due to perceptual aliasing. RPE cannot recover the true location from a retrieved wrong place. CoPR helps retrieve the correct place, thus solving errors that RPE cannot.

While the distance-based loss function helps to retain viewpoint information among descriptors, we observed that there is still room for improvement in comparison to retrieving the GT Euclidean closest reference descriptors in the physical space. Future works could investigate architectures and loss functions that further enforce the network to learn feature representations useful for retrieving the 3-D Euclidean closest match. As shown in this work, anchor selection and descriptor extrapolation are two separate steps for map densification. In the future, a separate treatment of both, i.e., learning good anchors and extrapolating well using multiple anchors, could lead to better map densification. We hope that this work helps to identify the important problem of map densification through CoPR for VPR and its relation to viewpoint variance, and motivates further research on improving VPR-based localization accuracy through CoPR.

REFERENCES

- [1] C. Toft et al., "Long-term visual localization revisited," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2074–2088, Apr. 2022.
- [2] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet, "A survey on visual-based localization: On the benefit of heterogeneous data," *Pattern Recognit.*, vol. 74, pp. 90–109, 2018.
- [3] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.
- [4] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DoF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2938–2946.
- [5] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 920–929.
- [6] S. Lowry et al., "Visual place recognition: A survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.
- [7] M. Zaffaret al., "VPR-Bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change," *Int. J. Comput. Vis.*, vol. 129, no. 7, pp. 2136–2174, 2021.
- [8] S. Garg, T. Fischer, and M. Milford, "Where is your place, visual place recognition?," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 2021, pp. 4416–4425.
- [9] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of CNN-based absolute camera pose regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3297–3307.
- [10] V. Balntas, S. Li, and V. Prisacariu, "RelocNet: Continuous metric learning relocalisation using neural nets," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 751–767.
- [11] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "CamNet: Coarse-to-fine retrieval for camera re-localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2871–2880.
- [12] T. Sattler et al., "Benchmarking 6DoF outdoor visual localization in changing conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8601–8610.
- [13] A. Moreau, N. Piasco, D. Tsishkou, B. Stanculescu, and A. de La Fortelle, "LENS: Localization enhanced by NeRF synthesis," in *Proc. Conf. Robot Learn.*, 2022, pp. 1347–1356.
- [14] Z. Chen et al., "Deep learning features at scale for visual place recognition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 3223–3230.
- [15] J. Revaud, J. Almazán, R. S. Rezende, and C. R. d. Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5106–5117.
- [16] J. Thoma, D. P. Paudel, A. Chhatkuli, and L. Van Gool, "Geometrically mappable image features," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2062–2069, 2020.
- [17] A. Torii et al., "Are large-scale 3D models really necessary for accurate visual localization?," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 814–829, Mar. 2021.
- [18] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3D point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 15–29.
- [19] L. Liu, H. Li, and Y. Dai, "Efficient global 2D–3D matching for camera localization in a large-scale 3D map," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2391–2400.
- [20] H. Taira et al., "InLoc: Indoor visual localization with dense matching and view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7199–7209.
- [21] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1744–1756, Sep. 2016.
- [22] E. Brachmann and C. Roth, "Learning less is more—6D camera localization via 3D surface regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4654–4662.
- [23] E. Brachmann et al., "DSAC—Differentiable RANSAC for camera localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2492–2500.
- [24] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4762–4769.
- [25] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6555–6564.
- [26] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1525–1530.
- [27] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6939–6946.
- [28] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, "VidLoc: A deep spatio-temporal model for 6-DoF video-clip relocalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2652–2660.
- [29] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 627–637.
- [30] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 870–877.
- [31] N. Radwan, A. Valada, and W. Burgard, "VLNet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 4407–4414, Oct. 2018.
- [32] S. Saha, G. Varma, and C. Jawahar, "Improved visual relocalization by discovering anchor points," in *Proc. Brit. Mach. Vis. Conf.*, 2018.
- [33] P.-E. Sarlin et al., "Back to the feature: Learning robust camera localization from pixels to pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3246–3256.
- [34] L. Yang, Z. Bai, C. Tang, H. Li, Y. Furukawa, and P. Tan, "SANet: Scene agnostic network for camera localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 42–51.
- [35] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *Int. J. Comput. Vis.*, vol. 124, no. 2, pp. 237–254, 2017.
- [36] F. Radenović, G. Toliás, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, Jul. 2019.
- [37] J. Thoma, D. P. Paudel, and L. Van Gool, "Soft contrastive learning for visual localization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 11119–11130.
- [38] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," in *Proc. Australian Conf. Robot. Automat.*, 2014, pp. 1–8.
- [39] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.
- [40] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of ConvNet features for place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 4297–4304.
- [41] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, "Generalized contrastive optimization of Siamese networks for place recognition," 2021, *arXiv:2103.06638*.
- [42] S. Hausler, A. Jacobson, and M. Milford, "Multi-process fusion: Visual place recognition using multiple image processing methods," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1924–1931, Apr. 2019.
- [43] S. Hausler and M. Milford, "Hierarchical multi-process fusion for visual place recognition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3327–3333.

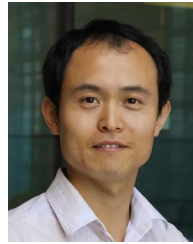
- [44] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4455–4465.
- [45] M. Oechsle, L. Mescheder, M. Niemeyer, T. Strauss, and A. Geiger, "Texture fields: Learning texture representations in function space," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4530–4539.
- [46] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3501–3512.
- [47] 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," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.
- [48] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "2d/7 place recognition by view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1808–1817.
- [49] Z. Zhang, T. Sattler, and D. Scaramuzza, "Reference pose generation for long-term visual localization via learned features and view synthesis," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 821–844, 2021.
- [50] 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. Intell. Robots Syst.*, 2021, pp. 1323–1330.
- [51] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time RGB-D camera relocalization," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2013, pp. 173–179.
- [52] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
- [53] E. Brachmann and C. Rother, "Expert sample consensus applied to camera re-localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7525–7534.
- [54] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12716–12725.
- [55] R. A. Newcombe et al., "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. IEEE 10th Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.
- [56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [57] G. Berton, C. Masone, V. Paolicelli, and B. Caputo, "Viewpoint invariant dense matching for visual geolocation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12149–12158.



Mubariz Zaffar received the bachelor's degree in electrical engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2016, and the master of science by dissertation degree in computer science and electronics engineering from the University of Essex, Colchester, U.K., in 2020. He is currently working toward the Ph.D. degree in computer vision and machine learning supervised by Dr. Julian Francisco Pieter Kooij and Dr. Liangliang Nan with the 3D Urban Understanding (3DUU) Lab, Delft University of Technology (TUD), Delft, The

Netherlands.

He is a member of the TUD Intelligent Vehicles Group headed by Prof. Dr. Dariu M. Gavrila. His research interests include place recognition, visual localization and mapping, and representation learning.



Liangliang Nan received the Ph.D. degree in mechatronics engineering from the Graduate University of the Chinese Academy of Sciences, Beijing, China, in 2009.

Before joining the Delft University of Technology as an Assistant Professor in 2018, he was a Research Scientist with the Visual Computing Center, King Abdullah University of Science and Technology. His research interests include computer vision, computer graphics, 3-D geoinformation, and machine learning.



Julian Francisco Pieter Kooij received the Ph.D. degree in visual detection and path prediction for vulnerable road users from the University of Amsterdam, Amsterdam, The Netherlands, in 2015.

Afterward, he joined the Delft University of Technology, first with the Computer Vision Lab, and later with the Intelligent Vehicles Group, where he is currently an Associate Professor. His research interests include deep representation learning and probabilistic models for multisensor localization, object detection, and forecasting of urban traffic.