MeshVPR: Citywide Visual Place Recognition Using 3D Meshes

MeshVPR: Citywide Visual Place Recognition Using 3D Meshes

Gabriele Berton[1], Lorenz Junglas[2], Riccardo Zaccone[1], Thomas Pollok[3], Barbara Caputo[1], and Carlo Masone[1]

[1] Politecnico di Torino
[2] Karlsruhe Institute of Technology
[3] Fraunhofer IOSB

Abstract. Mesh-based scene representation offers a promising direction for simplifying large-scale hierarchical visual localization pipelines, combining a visual place recognition step based on global features (retrieval) and a visual localization step based on local features. While existing work demonstrates the viability of meshes for visual localization, the impact of using synthetic databases rendered from them in visual place recognition remains largely unexplored. In this work we investigate using dense 3D textured meshes for large-scale Visual Place Recognition (VPR). We identify a significant performance drop when using synthetic mesh-based image databases compared to real-world images for retrieval. To address this, we propose MeshVPR, a novel VPR pipeline that utilizes a lightweight features alignment framework to bridge the gap between real-world and synthetic domains. MeshVPR leverages pre-trained VPR models and is efficient and scalable for city-wide deployments. We introduce novel datasets with freely available 3D meshes and manually collected queries from Berlin, Paris, and Melbourne. Extensive evaluations demonstrate that MeshVPR achieves competitive performance with standard VPR pipelines, paving the way for mesh-based localization systems. Data, code, and interactive visualizations are available at `https://meshvpr.github.io/`

Keywords: Visual Place Recognition (VPR) · 3D City Meshes · Image Retrieval

1 Introduction

Estimating the location of where a photo was taken based solely on its visual content is a staple of computer vision, and enables a number of applications ranging from augmented reality [45], robotics localization [56] and assistive technology [13]. It can be used as an alternative to GPS, where no signal nor internet connection is available or jammed [12, 68]. Additionally, it can help with automatic localization of non-geotagged image and video footage, which can be useful during or after tragic events like the terror attacks of Paris in 2015, to enable a time-critical investigation instead of months of manual labour.

The approaches developed to tackle the problem depend on factors like the size of the localization map (*e.g.* a small building vs a large city) and the required accuracy of the estimate (*e.g.*, a coarse position or a precise pose). In the most challenging scenarios, *i.e.*, when the objective is the precise pose on a large map, it is common practice to rely on hierarchical solutions [18, 36, 47, 49, 57, 59, 63], which comprise two steps: (1) a *Visual Place Recognition* (**VPR**) step with global features, where methods such as NetVLAD [5] are used to obtain a coarse prediction through efficient image retrieval, and (2) a *Visual Localization* (**VL**) step with local features [48], where the initial pose estimate is refined, often by matching keypoints from the image to be localized to a 3D model of the map.

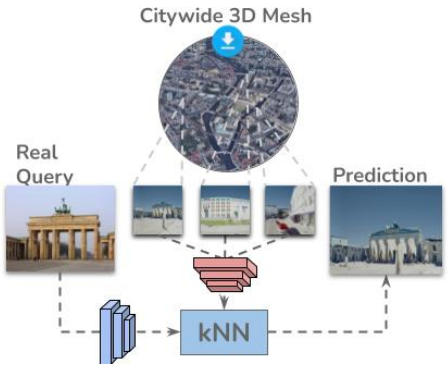

Fig. 1: Mesh-based VPR. Deploying a VPR model to a city can be performed with a database of synthetically generated images from a 3D mesh. While standard image retrieval techniques can be applied, a natural drop in results is due to the domain shift between real images (the query to be localized) and the synthetic database, requiring for the search of new solutions.

Both VPR and VL have been thoroughly studied in literature, either independently or within hierarchical pipelines. The vast majority of works relies on creating a world map with RGB images, which are used in VPR as a retrieval database, and can then be employed to create a 3D point cloud of the map, with each point being associated to a feature vector. Such point clouds are inherently tied to the model used to extract their features, limiting their flexibility and adaptability to new methods.

Using a different approach, a separate branch of literature demonstrates the viability of using dense 3D textured mesh models as scene representation for visual localization [6, 36, 37, 52, 53, 70]. Overall, such methods find that the synthetic-to-real domain shift between the map and queries does not strongly impact the localization, noting that local features are robust to such visual changes. Conversely, this synthetic-to-real shift has been scarcely investigated on global features for VPR: previous mesh-based localization works have either (i) used small scenes, skipping the VPR step altogether [6, 52, 53], (ii) relied on a database of real images for retrieval, before a mesh-based post-processing [36], or (iii) used retrieval on maps of limited dimension [37, 65] and did not report a noticeable performance drop related to the domain shift.

Addressing this knowledge gap is crucial for enabling the creation of fully mesh-based large-scale hierarchical localization pipelines – an attractive direction due to the increasing availability of free 3D textured meshes for multiple cities [23]. This work is the first investigation on VPR using global features on citywide 3D meshes: on a large-scale San Francisco dataset we find that a SOTA

VPR model achieves a Recall@1 of 96.3% when using a real-images database, and drops to 76.9% when using a database made of synthetic images (see Fig. 1).

To bridge this gap, we propose a novel VPR model that excels in three key aspects: (i) it efficiently matches real-world photos to a database of synthetic images, (ii) it scales effectively for large-scale datasets, enabling city-wide deployments, and (iii) it delivers strong results, achieving performance competitive with standard real-world VPR pipelines. We achieve this by developing a lightweight features alignment procedure that leverages pre-trained, state-of-the-art VPR models and adapts them to ensure consistent feature representations between real and synthetic domains. This paper details the complete pipeline, including training data preparation, features alignment, testing data acquisition from 3D meshes, and model deployment. We then evaluate our pipeline over a number of novel datasets, which rely on freely available 3D meshes and manually collected sets of queries. Finally, we demonstrate how our pipeline, called MeshVPR, achieves impressive results on each of the datasets, performing competitively with standard VPR pipelines using real-world database images. To summarize, our contributions are:

- MeshVPR, a novel mesh-based VPR pipeline that uses a lightweight, model-agnostic and hyper-parameters free features alignment step to compensate for the mismatch between real-world and generated images. MeshVPR leverages the broad availability of citywide 3D meshes as well as pretrained VPR models to enable quick deployment of a VPR system on a target city.
- The release of our three test sets from Berlin, Paris, and Melbourne, with manually collected real-world queries. Additionally, we also release a set of synthetic images from San Francisco to be used together with the already existing SF-XL [8] dataset to perform features alignment, without the need to collect more real-world images.
- A thorough analysis of failure cases and open challenges, noting that despite the large domain gap MeshVPR is able to achieve excellent quantitative and qualitative results, leading the way for future research and for new mesh-based localization systems.

2 Related work

Hierarchical Visual Localization pipelines [46, 62] rely on the combination of two steps: a visual place recognition (retrieval) step to obtain a coarse pose estimate, and a visual localization step, which aims at obtaining a precise camera pose. While both tasks have been largely investigated for real-to-real localization, and recent works explored the possibility of mesh-based visual localization [36, 37], mesh-based visual place recognition is still largely unstudied, despite being crucial for scalable hierarchical pipelines. Below we present a summary on (real-to-real) visual place recognition and syn-to-real localization works.

Visual Place Recognition Early VPR methods relied on representations obtained from hand-crafted local features, such as SIFT [27] and RootSIFT [4],

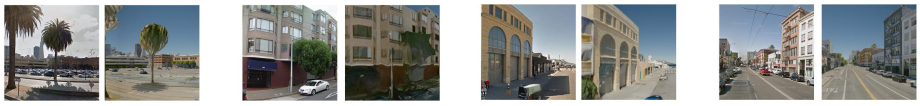

Fig. 2: Pairs of real images and their synthetic counterpart. Pairs like these are used for MeshVPR's features alignment.

while modern approaches have shifted to deep learning based approaches, where features are learned with CNNs [5, 7, 50, 64] or visual transformer architectures [71]. Several works have proposed methods to condense these features into compact and discriminative global embeddings, *e.g.* using pooling layers [5, 17, 30, 34, 43, 44, 60], clustering based approaches [21, 38, 69], MLPs [3] or self-attention [71]. Most notably, to ensure that the model learns to extract specific features for urban VPR, Arandjelovic et al. [5] proposed to train it on a dataset of StreetView images in a weakly supervised way. Many following works built on top of NetVLAD, enhancing it with an attention module [21], a novel loss [26], or a self-supervised training strategy [16]. Recent state-of-the-art methods [3, 10, 19] rely on efficient training paradigms, and leverage large-scale training datasets such as GSV-cities [2] and SF-XL [8].

Synthetic data for localization tasks The use of synthetic images has been explored for pose estimation with respect to a large indoor 3D map [58]. Aiming at addressing changes in appearances, [61] has explored view synthesis to align viewpoints of queries and database. Despite the existence of numerous synthetic 3D meshes of cities [14, 23] and synthetically-generated environments [1, 15, 24, 25, 41, 42, 51], their usage for image localization has been little explored in recent years. Among the few outliers, Panek et al [36, 37] proposes the use of 3D meshes for visual localization, although still relying on real-world images for the retrieval step. On a separate development, Vallone et al [65] spearheaded the exploration of aerial images to render street-view images for place recognition, and used it to generate a database of 44k images for the coarse localization of real-world query images. More in general, [33] explored the use of synthetic data to improve the localization process, while others further train the model with real and synthetic data to improve feature representation [32].

This paper fills a noticeable gap in the literature, as it is the first work to (i) explore 3D meshes for citywide VPR, (ii) quantify the gap in results between using real images and synthetic images, (iii) use features alignment to efficiently adopt robust SOTA VPR models and (iv) provide a number of large-scale datasets comprising challenging manually-collected query images.

3 Mesh-based Visual Place Recognition Pipeline

From standard to mesh-based VPR. VPR systems are usually implemented following an image retrieval approach. The images of the target city's database constitute the hypotheses for the incoming visual query, and predicting its loca-

tion involves a similarity search in that database. In practice, the query's location is inferred from the most relevant matches of this retrieval process. Formally, the database $(\mathcal{X}, \mathcal{Y})$ is composed of images $x \in \mathcal{X}$ annotated with a label $y \in \mathcal{Y}$ from the target city where the system needs to be deployed. This pipeline consists of two steps [29]: first, the query image x_q to be localized and all the images from the database $\mathcal{X}$ are mapped to a common embedding space determined by a model f_θ with learnable parameters θ; then, a k-nearest neighbor search (kNN) is performed to retrieve the database images most similar to the query x_q, and their labels.

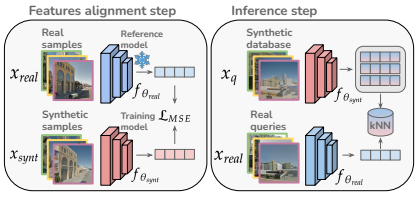

Fig. 3: At training time the goal is to align features extracted by $f_{\theta_{real}}$ from real images to those extracted by $f_{\theta_{synt}}$ from synthetic images. **At testing time** features extracted from synthetic images are used to localize the query (a real image).

Motivated by the ever-growing availability of freely-available city-wide 3D textured mesh models [23], and the recent advances in mesh-based visual localization [36, 37], our idea is to use such 3D meshes to obtain a synthetic database. Note that these meshes are generally built from aerial images, which can not directly be used for VPR. Formally, given a target city where we want to deploy a VPR system, we define 3D mesh as $\mathcal{M}_{target}$, and the database of images extracted from it as $(\mathcal{X}^{synt}_{target}, \mathcal{Y}_{target})$, the goal is to correctly match a given query image to a synthetic image from $\mathcal{X}^{synt}_{target}$ that depicts the same scene. To allow VPR models to perform well in such conditions (*i.e.* real query against a synthetic database), we devised a new *mesh-based VPR pipeline* that can be applied to any existing VPR model $f_{\theta_{real}}$, trained on real-world data. Due to the appearance gap between real and synthetic images (see Fig. 2), the features that $f_{\theta_{real}}$ extracts from a couple of real/synthetic images, even from the same viewpoint, are not well aligned. To address this problem, we use a new model $f_{\theta_{synt}}$, that is initialized by setting $\theta_{synt} = \theta_{real}$ and then fine-tuned so that the embeddings $f_{\theta_{synt}}(x_{synt})$ and $f_{\theta_{real}}(x_{real})$ for a pair of matching synthetic/real images (*i.e.*, from the same viewpoint) are well aligned. This is akin to a teacher-student paradigm, where $f_{\theta_{real}}$ is the teacher and $f_{\theta_{synt}}$, although this differs with many teacher-student models in that (i) the two models have the same architecture and (ii) they are trained on images from different domains that share the exact same viewpoint. Finally, the deployment is completed by using the specialized model $f_{\theta_{synt}}$ to extract the embeddings from the synthetic database $\mathcal{X}^{synt}_{target}$, whereas the model $f_{\theta_{real}}$ extracts the embeddings from the real-world queries. This idea is illustrated in Fig. 3 (right). The overall deployment pipeline is a *5-steps recipe*, as illustrated in Fig. 4. Remarkably, the features alignment (steps 1-3) needs to be performed only once to align the model $f_{\theta_{synt}}$, while any actual deployment only requires the database extraction from the 3D mesh of the target city.

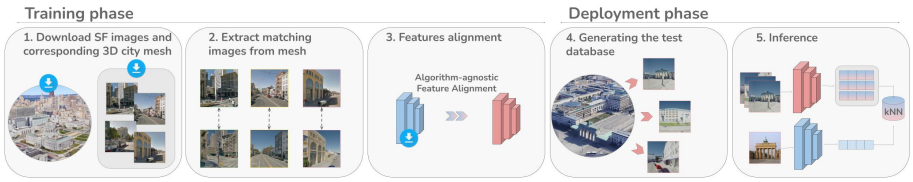

Fig. 4: Our proposed pipeline for mesh-based visual place recognition. The training phase consists in downloading training (real) images and the 3D mesh, generating their synthetic counterparts and specializing the synthetic model through feature alignment. Once the training phase is completed, the deployment phase can take part on any target city: in this paper we show results on Berlin, Paris and Melbourne.

3.1 Step 1: Download images and mesh for the alignment

The crucial part in our pipeline is the features alignment (described in Sec. 3.3), such that real and synthetic images are aligned in the feature space. To prepare for this, it is necessary to download a 3D mesh for any city for which a training set of real images is available. Notably, there is no need for this dataset and 3D mesh to be from the target city(ies), and in most of our experiments we pick a different city from the target one. Therefore, in this step, we simply download a dataset of real-world images $\mathcal{X}^{real}_{align}$ and a 3D mesh $\mathcal{M}_{align}$ from the same city. The images $\mathcal{X}^{real}_{align}$ must be labeled both with GPS coordinates and heading, since orientation is crucial in the next step to generate synthetic images from the same viewpoint as the real ones.

3.2 Step 2: Generate alignment images from mesh

With downloaded the citywide 3D mesh and the corresponding training dataset, it is then possible to create a set of synthetic images $\mathcal{X}^{synt}_{align}$ from the mesh $\mathcal{M}_{align}$, such that they precisely mirror the images from $\mathcal{X}^{real}_{align}$ (see Fig. 2).

This requires to have the latitude, longitude and heading (yaw) of each real image in $\mathcal{X}^{real}_{align}$, as they are needed to create views of the mesh from the corresponding viewpoints. Altitude, pitch and roll are also required to accurately replicate the images, but these can be inferred from the physics information of the 3D mesh. To accomplish that, we cast a ray from the real image location towards the ground and see where it intersects with the model. This intersection point serves as a reliable estimate of the ground level. The views are then generated at a height of 2.5m, *i.e.*, roughly the height of a typical car-mounted camera used to collect VPR datasets [28, 31]. Additionally, the normal vector at the intersection is used to estimate pitch and roll.

Synthetic counterpart for real images can be generated from the mesh using any rendering engine: in our case, we rely on Unreal Engine and Cesium.

3.3 Step 3: Features alignment

In order to extract coherent embeddings from both real-world and synthetic images, we propose a strategy based on two expert models, $f_{\theta_{real}}$ and $f_{\theta_{synt}}$, for

real and synthetic images respectively. The goal of this step is to ensure that $f_{\theta_{synt}}$ and $f_{\theta_{real}}$ produce similar embeddings from images that are taken from the same viewpoint.

To this end, we initialize $f_{\theta_{real}} = f_{\theta_{synt}}$ to any open-source pretrained VPR model, like CosPlace [8], MixVPR [3] or SALAD [19]. Secondly, we fine-tune $f_{\theta_{synt}}$ on the images $\mathcal{X}^{synt}_{align}$ to mirror the features extracted by $f_{\theta_{real}}$ from the corresponding images in $\mathcal{X}^{real}_{align}$. Formally, we optimize the following MSE loss:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(f_{\theta_{real}}(x_{real,i}) - f_{\theta_{synt}}(x_{synt,i})\right)^2 \tag{1}$$

where $N = |\mathcal{X}^{real}_{align}| = |\mathcal{X}^{synt}_{align}|$ and $x^{real}_i \in \mathcal{X}^{real}_{align}$, $x^{synt}_i \in \mathcal{X}^{synt}_{align}$ are pairs of real and synthetic images from the same viewpoint, and the parameters θ_{real} are frozen.

Note that: (i) the training phase (steps 1-3) needs to be performed only once to create the model $f(\theta_{synt})$, which can then be deployed to any new city, and (ii) although these steps do require a real-world dataset $\mathcal{X}^{real}_{align}$ annotated with position and heading, it is possible to use existing datasets such as SF-XL [8] without having to rely on images from the target city.

3.4 Step 4: Generate the test database

The next step consists in generating the database $(\mathcal{X}^{synt}_{target}, \mathcal{Y}_{target})$ for the target cities, *i.e.*, the ones where we want to deploy the VPR system. Given the 3D mesh $\mathcal{M}_{target}$ from the target city, we want to ensure that the database is constructed by generating relevant views, which could ideally match the camera viewpoints of the queries.

We generate street-view-like images from the 3D mesh by simulating the path that a camera-equipped car would take to collect a database. This process begins with fetching OpenStreetMap (OSM) street data for the area covered by the 3D mesh $\mathcal{M}_{target}$. Then, a graph is constructed from the OSM data. For this graph we compute a path that visits every edge, this is accomplished by solving the Route Inspection Problem for the graph. This continuous path is needed in order to ensure equal spacing between the individual sampling locations and reduce the sampling time. The actual capture is performed by moving the camera along the path by a configurable distance (10 meters in our experiments, unless otherwise stated). We note that sampling can easily be adapted to the specific application, *e.g.* a database for drone localization can be sampled from simulated aerial viewpoints. This underscores the flexibility of our approach for scenarios that are different from street-level visual place recognition.

3.5 Step 5: Inference

Finally, we deploy the system using the generated test database $(\mathcal{X}^{synt}_{target}, \mathcal{Y}_{target})$, and the models $f_{\theta_{real}}$ and $f_{\theta_{synt}}$. First, we extract the embeddings from each

Table 1: Characteristics of 3D mesh models used in our experiments. The 5 central columns refer to the 3D mesh. The 3 right-most columns refer to the images used, either generated from the mesh (for the database) or the queries. San Francisco HQ and San Francisco LQ indicate the higher/lower quality ones, which we use to investigate how mesh resolution affects results. DB stands for database. *Only the subset of San Francisco XL (SF-XL) overlapping both 3D meshes is considered; †Different parts of the mesh have different quality.

Dataset Type	City	Size (GB)	Size (sq km)	Resolution (cm/px)	Year	Provider	DB images (GB)	# DB images	# queries
Train sets	San Francisco XL (real images)	-	-	-	-	-	55	9.2M	134
	San Francisco HQ	36	16.4	0.6; 2; 5 †	2021	Aerometrex	55	9.2M*	134*
	San Francisco LQ	8	16.4	≈ 15	2015-2022	Google	55	9.2M	134
Test sets	Berlin	71	22.7	≈ 9	2020	Senate of Berlin	41	1.3M	255
	Paris	11	25.3	≈ 15	2015-2023	Google	45	1.8M	268
	Melbourne	21	13.4	7.5	2018	City of Melbourne	11	394k	1249

image in $\mathcal{X}^{synt}_{target}$ using $f_{\theta_{synt}}$. Then, the inference simply consists in taking any unseen real-world query x_q, extracting its embedding $f_{\theta_{real}}(x_q)$, and compare it to the embeddings of the database images via kNN. Once the nearest neighbor(s) is found, we can directly infer the query's position from its metadata, following standard visual place recognition [5].

4 Test sets

To empirically validate the soundness of MeshVPR, we built three test sets, each one consisting of a synthetic geo-tagged database, built following the method in Sec. 3.4, and a number of real-world queries. The three test sets cover the cities of Berlin, Paris and Melbourne, for which 3D meshes are freely available: the datasets characteristics are summarized in Tab. 1. For Berlin and Paris, a part of the queries have been manually collected, and another part has been downloaded from Flickr and Wikimedia, which are commonly used sources for computer vision datasets [8, 39, 40]. Some examples are shown in the qualitative results in Sec. 5 and in the Supplementary. Given the inaccuracies in GPS positioning (both in our manually collected queries, and Flickr and Wikimedia photos), we manually selected only a few hundred queries for the two European capitals, carefully removing any image for which the GPS label did not match its actual position. For Melbourne we use as queries a subset of the Mapillary Street-Level Sequences dataset [67] collected in the city of Melbourne.

5 Experiments

In this section we present quantitative and qualitative results on mesh-based VPR, starting with MeshVPR's implementation details (Sec. 5.1), analyzing results on mesh-based visual place recognition with standard VPR models with and without the integration of MeshVPR (Sec. 5.2), analyzing the importance of mesh quality for the task (Sec. 5.3), exploring the performance gap when using a real or a synthetic database (Sec. 5.4), exploring strategies alternative to

Table 2: Evaluating methods on mesh-based VPR. For each method, we show their performance with and without MeshVPR on our three new datasets made of synthetic database and real queries.

Method	Synt-Berlin					Synt-Paris					Synt-Melbourne				
	R@1	R@5	R@10	R@20	R@100	R@1	R@5	R@10	R@20	R@100	R@1	R@5	R@10	R@20	R@100
Conv-AP	37.3	49.4	56.1	60.8	75.7	34.3	47.8	55.2	61.2	73.5	18.7	30.7	39.3	49.7	75.6
Conv-AP + MeshVPR (Ours)	41.6	57.6	62.7	71.4	84.3	35.4	51.5	56.3	62.7	76.1	25.0	43.0	51.9	60.7	82.5
CosPlace	55.7	64.7	69.4	72.5	82.4	49.6	59.0	61.9	65.3	76.1	38.4	52.1	61.5	69.5	84.9
CosPlace + MeshVPR (Ours)	63.9	74.1	80.0	85.9	92.2	50.7	62.3	68.3	73.1	79.9	49.5	63.4	72.1	78.1	91.2
MixVPR	48.6	63.1	67.5	72.5	83.9	45.1	56.7	59.3	66.4	76.5	28.4	45.6	53.1	61.0	81.8
MixVPR + MeshVPR (Ours)	60.4	73.3	79.2	82.4	91.0	52.2	60.4	66.0	73.1	82.1	38.9	53.5	60.8	68.4	89.5
EigenPlaces	52.2	63.5	67.8	72.9	79.6	44.8	56.0	61.9	66.0	73.5	26.7	40.4	47.6	54.2	74.7
EigenPlaces + MeshVPR (Ours)	67.1	80.4	83.5	85.9	92.2	48.5	59.3	64.6	69.8	81.0	47.6	64.1	70.3	75.6	89.6
SALAD	69.8	79.6	83.1	84.7	93.3	53.7	68.3	71.3	74.6	83.6	59.1	73.9	78.8	84.3	94.6
SALAD + MeshVPR (Ours)	**82.0**	**89.4**	**92.2**	**92.9**	**95.7**	**63.8**	**76.5**	**80.2**	**81.3**	**84.7**	**69.1**	**81.7**	**86.8**	**92.2**	**97.8**

MeshVPR (Sec. 5.5), and finally analyzing advantages and limitations of using real or synthetic data for image localization.

5.1 Implementation details

Features Alignment. Our feature alignment is performed by pairing two copies of the same VPR model (*e.g.*, NetVLAD, CosPlace, MixVPR), both initialized with open-source pretrained weights. One of the two models, which we refer to as $f_{\theta_{real}}$, has frozen weights θ_{real}, while the weights θ_{synt} of $f_{\theta_{synt}}$ are fine-tuned for 50k iterations with a batch size of 32 and the Adam [22] optimizer with learning rate $1e-5$. Training takes only 3 hours on just 4GB of VRAM when using the heaviest model considered (*i.e.* SALAD) on a Nvidia RTX-4090 GPU.

As per the training set, we use the largest publicly available VPR dataset, namely San Francisco eXtra Large (SF-XL) [8] and its 3D mesh counterpart, with synthetic views extracted from the San Francisco HQ mesh (see Tab. 1) unless otherwise stated.

Inference. At inference time, $f_{\theta_{synt}}$ is used to extract features from synthetic database, while the queries are processed through $f_{\theta_{real}}$. As in standard VPR [2, 3, 5, 8, 10, 16], queries' features are matched against database ones by kNN algorithm (see Fig. 3, right). We use a threshold of 100 meters for positives.

5.2 Localizing Real Queries on a Synthetic Database

Firstly, we investigate the feasibility of localizing real-world queries through a synthetic database. To this end, we use a number of standard VPR methods, to see how results change when MeshVPR is applied. Note that MeshVPR can be applied on top of any VPR model, which we showcase by testing SOTA single-stage VPR methods since 2022 [2, 3, 8, 10, 19] with and without MeshVPR. As it is possible to notice in Tab. 2, even methods that have been proved to be robust to diverse training and test distributions, like MixVPR and EigenPlaces,

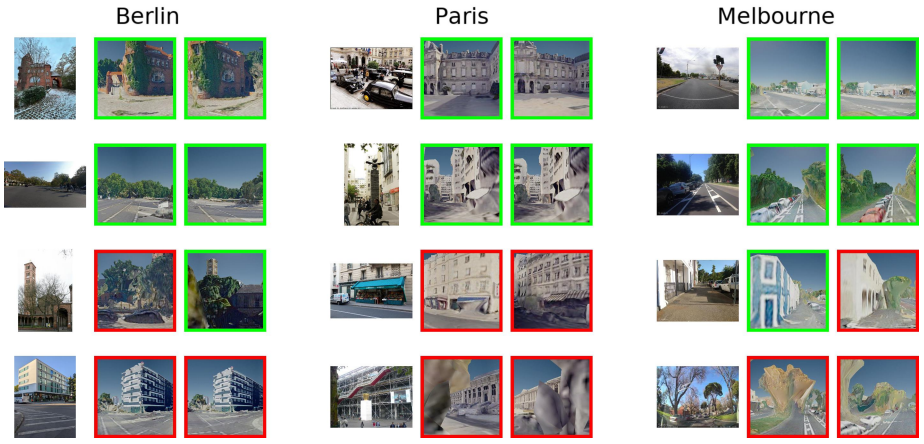

Fig. 5: Predictions with best MeshVPR model, namely SALAD + MeshVPR. Each triplet represents a query and its top 2 predictions, which are bounded in green if positive and red if negative. Qualitative examples help understand the results from Tab. 2: Paris is challenging due to low quality meshes, and Melbourne is challenging due to wide open spaces. Interestingly, we note that the model learns to overcome long-term temporal changes (snow and winter/summer foliage in top-left query), occlusions (first two queries from Paris) and perspective changes (third query from Melbourne). A large number of (higher resolution) qualitative results are shown in the Supplementary.

perform poorly in the mesh-based VPR setting, confirming that the challenges posed by using synthetic images are beyond the generalization capabilities of any specific algorithm. MeshVPR allows to achieve impressive results, greatly increasing the respective baselines. Qualitative results in Fig. 5 give an insight into what MeshVPR learns during its feature alignment stage: most notably, the model is able to overcome long-term temporal changes, seasonal changes, occlusions and perspective changes. We note also how it is able to match features that have highly different appearance within the two domains, for example queries with trees produce predictions with trees, despite real and synthetic trees having little visual similarity.

5.3 How 3D mesh quality affects results

Given that citywide 3D meshes play a central role in MeshVPR, we investigate the effect of their quality on the mesh-based VPR system. For this purpose, we take the two overlapping 3D mesh of San Francisco LQ and San Francisco HQ (cf. Tab. 1) and use them both to generate synthetic views of the city (cf. Fig. 6).

We provide results when the features alignment is performed on each of these datasets (whereas the real images for features alignment are unchanged). To ensure that the areas of features alignment and testing do not overlap, we split the datasets into two non-overlapping halves (respectively north and south

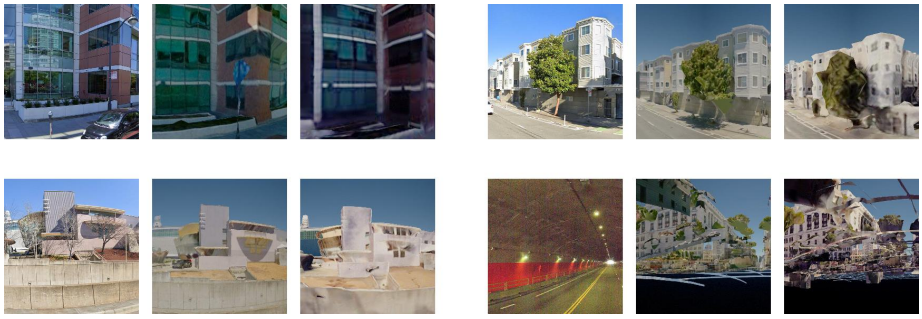

Fig. 6: Triplets of real, synthetic from HQ mesh, and synthetic from LQ mesh. These triplets allow to qualitatively understand how the quality of the mesh influences the generated images and results. The bottom-right triplet provides a examples of synthetic images with artifacts. They occur when the real image was taken in a covered area i.e tunnel or tree cover, and the viewpoint is *within* the mesh. Examples with such artifacts account for less than 1% of the dataset.

of latitude 37.78°) that are used as training and test sets, respectively. Note that we do not need validation as MeshVPR does not have hyperparameters, nor we perform early stopping. The choice of using geographically disjoint train/test sets is in accordance to the typical VPR approach [5, 9, 67], where non-overlapping train/test sets are used. As queries, we use those from SF-XL [8] that are within the test area.

Results are shown in Fig. 7, and they prove that the mesh quality has a huge impact on the results. Using the HQ (high quality) mesh (for both train and test) leads to a 15.7 points R@1 improvement over the LQ (low quality) mesh. Results show that the scores obtained when testing on the LQ database benefit if the training is also performed on LQ data. For this reason, we will release both the models trained with HQ and LQ data.

Albeit somewhat predictable, this result is promising for the development of mesh-based VPR: in fact, as the quality (and availability) of these city-wide 3D meshes is steadily increasing, so will the results of MeshVPR.

5.4 Bridging the Syn2Real performance gap

Extensive validation presented in the above section demonstrates that MeshVPR makes each VPR algorithm obtain satisfactory performances on the mesh-based VPR setting. A natural question arises: *to what extent do we recover the performance we would have obtained if a database of real images had been available?*

To answer this, we conducted an experiment on both real and synthetic versions of the San Francisco dataset, again splitting the datasets along latitude 37.78° in training and test sets, to ensure that MeshVPR's features alignment is not performed on the database samples. We compare the performance of state-of-the-art VPR models both when using a database of real-images, and when using the synthetic database. Results in Fig. 8 show that, while a gap does still

Method	Trained on	Tested on	R@1	R@10	R@100
MixVPR	LQ DB	LQ DB	59.7	74.6	91.0
		HQ DB	59.0	77.6	93.3
	HQ DB	LQ DB	53.0	69.4	91.8
		HQ DB	70.9	83.6	94.0
EigenPlaces	LQ DB	LQ DB	75.4	82.1	94.0
		HQ DB	80.6	88.1	94.0
	HQ DB	LQ DB	67.9	81.3	94.8
		HQ DB	85.8	91.8	97.0
SALAD	LQ DB	LQ DB	76.1	90.3	97.8
		HQ DB	87.3	91.0	97.8
	HQ DB	LQ DB	73.1	86.6	97.8
		HQ DB	88.8	94.8	97.8

Fig. 7: Results with MeshVPR on High Quality (HQ) and Low Quality (LQ) meshes. Quantitative results (left) indicate a strong correlation between results and mesh quality. All results on the table are computed with MeshVPR applied to different VPR models. Qualitative results (right) visually show how predictions are affected by the synthetically generated images. For each one of the 4 queries (*i.e.* the images without green/red boxes) we show the top-2 candidates with SALAD+MeshVPR on the high-quality (HQ) database (top row for each query) and the top-2 candidates with the low-quality (LQ) database.

exist, MeshVPR recovers much of the performance loss caused by the use of the synthetic database. Most notably, the best state-of-art VPR model (SALAD) experiences a drop in Recall@5 (R@5) $< 4\%$: the performance of SALAD with a real-world database is 97.8%, opposed to 94.0% of SALAD+MeshVPR on a synthetic database.

5.5 Comparing MeshVPR with other strategies

Results from Tab. 3 demonstrate the benefit of MeshVPR's two-models strategy, with one model extracting features from real images and the other extracting aligned features from synthetic images. On the other hand, other solutions only alleviate the problem and do not achieve competitive results with MeshVPR.

5.6 Training on fewer images

The San Francisco datasets (XL, HQ, and LQ) that we use for MeshVPR's features alignment all have exactly the same number of images from the same viewpoints, and comprise a large number of over 9.2M images each. Whereas so many images are necessary to provide a proper measure of how changing the database's domain (*i.e.* real or synthetic) for inference affects the results (see Fig. 8), in this section we aim at understanding if so many images are actually necessary for the features alignment. We report results in Tab. 4, where we can see that using a much smaller training set of 100 k images leads to competitive results to using the full dataset. Also, as detailed in Sec. 5.1, note that in practice

Scenario	Method	R@1	R@10	R@100
Standard VPR	Conv-AP	73.1	91.0	96.3
	CosPlace	88.1	93.3	98.5
	MixVPR	88.1	94.8	97.0
	EigenPlaces	92.5	96.3	98.5
	SALAD	96.3	98.5	99.3
Mesh-based VPR	Conv-AP	31.3	51.5	78.4
	CosPlace	67.2	79.9	88.8
	MixVPR	48.5	70.1	90.3
	EigenPlaces	64.9	79.9	89.6
	SALAD	76.9	88.8	97.8
	Conv-AP + MeshVPR (Ours)	51.5	74.6	93.3
	CosPlace + MeshVPR (Ours)	80.6	86.6	93.3
	MixVPR + MeshVPR (Ours)	70.9	83.6	94.0
	EigenPlaces + MeshVPR (Ours)	85.8	91.8	97.0
	SALAD + MeshVPR (Ours)	88.8	94.8	97.8

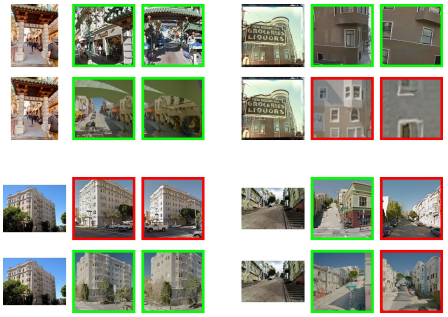

Fig. 8: Quantitative and qualitative results with a real vs a synthetic database. Quantitative results (left) show are performed with SOTA VPR models on a real-world database (top-5 rows), and mesh-based database with and without MeshVPR. Qualitative results (right) show examples of 4 queries with their top-2 predictions, with the prediction bounded in green/red if correct/wrong. For each query we show predictions with SALAD on the real DB (top), and SALAD+MeshVPR on the synthetic DB (bottom).

Table 3: Using metric learning losses to train a model on a dataset containing both synthetic and real images. All methods start from a pretrained ResNet50-based EigenPlaces model. Other methods apply a "standard" fine-tuning of the model (in this case using both synthetic and real images), whereas MeshVPR uses the features alignment.

Method	Synt-Berlin					Synt-Paris					Synt-Melbourne				
	R@1	R@5	R@10	R@20	R@100	R@1	R@5	R@10	R@20	R@100	R@1	R@5	R@10	R@20	R@100
Contrastive loss	35.9	46.2	52.1	55.7	68.6	32.8	44.0	48.3	52.8	66.2	19.8	29.4	34.8	41.7	61.5
Triplet loss as in [65]	34.1	44.0	48.8	54.7	68.6	32.0	43.3	47.5	53.6	64.9	20.3	31.0	37.2	43.8	64.0
Lifted loss [54]	38.0	48.5	53.7	59.1	73.3	34.5	45.6	50.8	55.2	65.2	22.6	35.1	41.8	47.8	67.9
NTXent loss [35]	37.6	48.5	54.5	59.0	72.3	34.1	47.1	50.9	55.6	66.6	23.4	35.5	41.4	49.2	68.0
FastAP loss [11]	37.3	48.2	54.2	60.1	72.4	34.2	45.8	50.6	54.6	66.5	24.5	37.6	43.9	50.1	68.6
MultiSimilarity [66]	40.4	50.6	57.8	63.3	77.5	37.4	47.8	52.4	57.1	68.5	24.5	35.5	41.7	49.4	69.1
Circle loss [55]	33.2	42.1	48.2	53.2	65.9	32.9	43.2	47.8	53.0	63.8	18.4	28.1	32.8	39.0	59.1
SupCon loss [20]	35.0	45.7	51.5	56.3	68.4	33.0	45.1	50.3	53.8	64.6	20.9	31.0	37.3	43.2	62.3
MeshVPR (Ours)	**67.1**	**80.4**	**83.5**	**85.9**	**92.2**	**48.5**	**59.3**	**64.6**	**69.8**	**81.0**	**47.6**	**64.1**	**70.3**	**75.6**	**89.6**

we train for 50k iterations with a batch size of 32, *i.e.* the model sees 1.6M images and never actually goes through the entire dataset (we found convergence to be fully reached by 50k iterations).

5.7 Limitations and advantages of mesh-based VPR

Standard VPR (with real images for a database) and mesh-based VPR come each with their own strengths and weaknesses. The main strength of standard VPR is the performance that SOTA models achieve even on large-scale datasets, as shown by the results in Fig. 8, (and [3, 10, 19]). Mesh-based VPR, having to overcome the strong domain shifts between database and query images, results

Table 4: Training on fewer images. Using the entire dataset (9.2 M pairs of images) for features alignment does not show large improvement, while using just 100 k images achieves almost the same results.

# training images	Synt-Berlin					Synt-Paris					Synt-Melbourne				
	R@1	R@5	R@10	R@20	R@100	R@1	R@5	R@10	R@20	R@100	R@1	R@5	R@10	R@20	R@100
10 k	77.6	86.3	89.4	91.8	96.1	59.3	72.8	75.7	81.0	84.3	67.3	82.8	87.0	92.6	98.2
100 k	81.2	88.6	90.2	92.2	95.7	61.2	76.1	78.4	79.5	84.7	70.5	82.1	87.3	92.3	97.9
1 M	80.0	87.5	90.2	93.3	96.1	60.1	73.5	77.6	81.0	85.1	69.7	80.6	85.6	90.9	97.8
9.2 M	82.0	89.4	92.2	92.9	95.7	63.8	76.5	80.2	81.3	84.7	69.1	81.7	86.8	92.2	97.8

in lower performance even with the strong improvements given by MeshVPR. On the other hand, mesh-based VPR provides a number of advantages:

1. Higher potential for collecting data in challenging scenarios, like dangerous or remote locations, inaccessible to cars because of factors like damaged infrastructure, but accessible to flying drones.
2. 3D citywide meshes provide a dense model of the city, covering virtually any outdoor location, whereas standard street-view based datasets cover only areas adjacent to roads.
3. Fewer privacy issues, due to features like faces and car plates being absent in the mesh. For real-world images, these can cause issues, and blurring is often required when using such images.
4. Increased flexibility in image sampling. 3D meshes allow simulating environmental factors, such as changes of appearance, light and weather conditions, which are difficult or expensive to capture in the real-world.

6 Conclusions and future works

In this work, we propose the task of citywide mesh-based visual place recognition, which aims at obtaining scalable solutions to the problem of image localization, relying solely on a large 3D mesh. We create three new datasets, to quantify the performance drop of existing models on this new task. We then develop a simple yet effective technique, called MeshVPR, to align two models to the same feature space – one model taking real images as inputs, and the other one taking synthetic images. MeshVPR can be trained on any dataset of real and synthetic image pairs, on top of any existing VPR model, showing impressive results and strong flexibility.

We believe that the novel problem setup introduced in this paper opens countless future directions. For example: (1) using a full mesh-based localization pipeline based on MeshVPR plus visual localization methods as in [36, 37], (2) generating synthetic images from multiple domains (e.g. synthetic night images), (3) simulate more viewpoints (e.g. from sidewalks or parks), (4) perform drone-based localization with a synthetic set of aerial images and (5) apply style transfer to transform all the images to a single domain. In conclusion, citywide mesh-based VPR is an exciting new line of research which could unleash its full potential in allowing ubiquitous localization in practical scenarios.

Acknowledgements. We acknowledge the Cineca award under the Iscra initiative, for the availability of high performance computing resources. This work was supported by CINI. Project supported by ESA Network of Resources Initiative. This study was carried out within the project FAIR - Future Artificial Intelligence Research - and received funding from the European Union Next-GenerationEU (Piano nazionale di ripresa e resilienza (PNRR) – missione 4 componente 2, investimento 1.3 – D.D. 1555 11/10/2022, PE00000013 - CUP: E13C22001800001). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. European Lighthouse on Secure and Safe AI – ELSA, Horizon EU Grant ID: 101070617

References

1. Alberti, E., Tavera, A., Masone, C., Caputo, B.: Idda: A large-scale multi-domain dataset for autonomous driving. IEEE Robotics and Automation Letters **5**(4), 5526–5533 (2020)
2. Ali-bey, A., Chaib-draa, B., Giguère, P.: GSV-Cities: Toward appropriate supervised visual place recognition. Neurocomputing **513**, 194–203 (2022)
3. Ali-bey, A., Chaib-draa, B., Giguère, P.: MixVPR: Feature mixing for visual place recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2998–3007 (2023)
4. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2911–2918 (2012)
5. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(6), 1437–1451 (2018). https://doi.org/10.1109/TPAMI.2017.2711011
6. Aubry, M., Russell, B., Sivic, J.: Visual Geo-localization of Non-photographic Depictions via 2D–3D Alignment, pp. 255–275. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-25781-5_14, https://doi.org/10.1007/978-3-319-25781-5_14
7. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: European Conference on Computer Vision. pp. 584–599. Springer International Publishing, Cham (2014)
8. Berton, G., Masone, C., Caputo, B.: Rethinking visual geo-localization for large-scale applications. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4868–4878 (2022). https://doi.org/10.1109/CVPR52688.2022.00483
9. Berton, G., Paolicelli, V., Masone, C., Caputo, B.: Adaptive-attentive geolocalization from few queries: A hybrid approach. In: IEEE Winter Conference on Applications of Computer Vision. pp. 2918–2927 (January 2021)
10. Berton, G., Trivigno, G., Caputo, B., Masone, C.: Eigenplaces: Training viewpoint robust models for visual place recognition. In: IEEE International Conference on Computer Vision. pp. 11080–11090 (October 2023)
11. Cakir, F., He, K., Xia, X., Kulis, B., Sclaroff, S.: Deep metric learning to rank. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1861–1870 (2019). https://doi.org/10.1109/CVPR.2019.00196
12. Chai, X., Yang, J., Yan, X., Di, C., Ye, T.: Efficient underground tunnel place recognition algorithm based on farthest point subsampling and dual-attention transformer. Sensors **23**(22) (2023). https://doi.org/10.3390/s23229261

13. Cheng, R., Hu, W., Chen, H., Fang, Y., Wang, K., Xu, Z., Bai, J.: Hierarchical visual localization for visually impaired people using multimodal images. Expert Systems with Applications **165**, 113743 (2021). https://doi.org/https://doi.org/10.1016/j.eswa.2020.113743, https://www.sciencedirect.com/science/article/pii/S0957417420305674
14. Deschaud, J.E., Duque, D., Richa, J.P., Velasco-Forero, S., Marcotegui, B., Goulette, F.: Paris-CARLA-3D: A real and synthetic outdoor point cloud dataset for challenging tasks in 3D mapping. Remote Sensing **13**(22) (2021). https://doi.org/10.3390/rs13224713, https://www.mdpi.com/2072-4292/13/22/4713
15. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. pp. 1–16 (2017)
16. Ge, Y., Wang, H., Zhu, F., Zhao, R., Li, H.: Self-supervising fine-grained region similarities for large-scale image localization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) European Conference on Computer Vision. pp. 369–386. Springer International Publishing, Cham (2020)
17. Hausler, S., Jacobson, A., Milford, M.: Multi-process fusion: Visual place recognition using multiple image processing methods. IEEE Robotics and Automation Letters **4**(2), 1924–1931 (2019)
18. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2599–2606. IEEE (2009)
19. Izquierdo, S., Civera, J.: Optimal transport aggregation for visual place recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (June 2024)
20. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Conference on Neural Information Processing Systems (2020), https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html
21. Kim, H.J., Dunn, E., Frahm, J.M.: Learned contextual feature reweighting for image geo-localization. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3251–3260 (2017)
22. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (12 2015)
23. Lei, B., Stouffs, R., Biljecki, F.: Assessing and benchmarking 3D city models. International Journal of Geographical Information Science **37**(4), 788–809 (2023). https://doi.org/10.1080/13658816.2022.2140808
24. Li, Y., Jiang, L., Xu, L., Xiangli, Y., Wang, Z., Lin, D., Dai, B.: MatrixCity: A large-scale city dataset for city-scale neural rendering and beyond. In: IEEE International Conference on Computer Vision. pp. 3205–3215 (October 2023)
25. Lin, C.H., Lee, H.Y., Menapace, W., Chai, M., Siarohin, A., Yang, M.H., Tulyakov, S.: Infinicity: Infinite-scale city synthesis. In: IEEE International Conference on Computer Vision. pp. 22808–22818 (October 2023)
26. Liu, L., Li, H., Dai, Y.: Stochastic attraction-repulsion embedding for large scale image localization. In: IEEE International Conference on Computer Vision. pp. 2570–2579 (2019). https://doi.org/10.1109/ICCV.2019.00266
27. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004), http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

28. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 Year, 1000km: The Oxford RobotCar Dataset. The International Journal of Robotics Research **36**(1), 3–15 (2017). https://doi.org/10.1177/0278364916679498
29. Masone, C., Caputo, B.: A survey on deep visual place recognition. IEEE Access **9**, 19516–19547 (2021). https://doi.org/10.1109/ACCESS.2021.3054937
30. Mereu, R., Trivigno, G., Berton, G., Masone, C., Caputo, B.: Learning sequential descriptors for sequence-based visual place recognition. IEEE Robotics and Automation Letters **7**(4), 10383–10390 (2022). https://doi.org/10.1109/LRA.2022.3194310
31. Milford, M., Wyeth, G.: Mapping a suburb with a single camera using a biologically inspired slam system. IEEE Transactions on Robotics **24**, 1038–1053 (2008)
32. Moreau, A., Piasco, N., Bennehar, M., Tsishkou, D.V., Stanciulescu, B., de La Fortelle, A.: Crossfire: Camera relocalization on self-supervised features from an implicit representation. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 252–262 (2023), https://api.semanticscholar.org/CorpusID:257427144
33. Moreau, A., Piasco, N., Tsishkou, D.V., Stanciulescu, B., de La Fortelle, A.: Lens: Localization enhanced by nerf synthesis. In: Conference on Robot Learning (2021), https://api.semanticscholar.org/CorpusID:238744321
34. Neubert, P., Schubert, S.: Hyperdimensional computing as a framework for systematic aggregation of image descriptors. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 16938–16947 (June 2021)
35. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. CoRR **abs/1807.03748** (2018), http://arxiv.org/abs/1807.03748
36. Panek, V., Kukelova, Z., Sattler, T.: MeshLoc: Mesh-based visual localization. In: European Conference on Computer Vision. pp. 589–609. Springer Nature Switzerland, Cham (2022)
37. Panek, V., Kukelova, Z., Sattler, T.: Visual localization using imperfect 3d models from the internet. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 13175–13186 (2023)
38. Peng, G., Zhang, J., Li, H., Wang, D.: Attentional pyramid pooling of salient visual residuals for place recognition. In: IEEE International Conference on Computer Vision. pp. 885–894 (October 2021)
39. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2007). https://doi.org/10.1109/CVPR.2007.383172
40. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2008). https://doi.org/10.1109/CVPR.2008.4587635
41. Pollok, T., Junglas, L., Ruf, B., Schumann, A.: Unrealgt: Using unreal engine to generate ground truth datasets. In: International Symposium on Visual Computing. p. 670–682. Springer (2019). https://doi.org/10.1007/978-3-030-33720-9_52
42. Qiu, W., Zhong, F., Zhang, Y., Qiao, S., Xiao, Z., Kim, T.S., Wang, Y.: UnrealCV: Virtual Worlds for Computer Vision. In: ACM MM. pp. 1221–1224. ACM (2017). https://doi.org/10.1145/3123266.3129396, https://dl.acm.org/doi/10.1145/3123266.3129396

43. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(7), 1655–1668 (2019). https://doi.org/10.1109/TPAMI.2018.2846566
44. Razavian, A.S., Sullivan, J., Carlsson, S., Maki, A.: Visual Instance Retrieval with Deep Convolutional Networks. ITE Transactions on Media Technology and Applications **4**(3), 251–258 (2016). https://doi.org/10.3169/mta.4.251
45. Sallam Fatouh, W., Farouk Ali, H., Abd Elrazek Mashali, S., Shouki Seliem, A.: Image-based localization for augmented reality application: A review. In: Proceedings of the 2021 5th International Conference on Virtual and Augmented Reality Simulations. p. 7–16. ICVARS '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463914.3463916, https://doi.org/10.1145/3463914.3463916
46. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 12716–12725 (2019)
47. Sarlin, P.E., Debraine, F., Dymczyk, M., Siegwart, R., Cadena, C.: Leveraging deep visual descriptors for hierarchical efficient localization. In: Conference on Robot Learning. pp. 456–465. Proceeding of Machine Learning Research (2018)
48. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning feature matching with graph neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2020), https://arxiv.org/abs/1911.11763
49. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image retrieval for image-based localization revisited. In: British Machine Vision Conference. vol. 1, p. 4 (2012)
50. Schubert, S., Neubert, P., Garg, S., Milford, M., Fischer, T.: Visual place recognition: A tutorial. IEEE Robotics & Automation Magazine pp. 2–16 (2023). https://doi.org/10.1109/MRA.2023.3310859
51. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In: Hutter, M., Siegwart, R. (eds.) Field and Service Robotics, Springer Proceedings in Advanced Robotics, vol. 5, pp. 621–635. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-67361-5_40, http://link.springer.com/10.1007/978-3-319-67361-5_40
52. Shan, Q., Wu, C., Curless, B., Furukawa, Y., Hernandez, C., Seitz, S.M.: Accurate geo-registration by ground-to-aerial image matching. In: International Conference on 3D Vision (3DV). vol. 1, pp. 525–532 (2014). https://doi.org/10.1109/3DV.2014.69
53. Sibbing, D., Sattler, T., Leibe, B., Kobbelt, L.: Sift-realistic rendering. In: International Conference on 3D Vision (3DV). pp. 56–63 (2013). https://doi.org/10.1109/3DV.2013.16
54. Song, H.O., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4004–4012 (2016). https://doi.org/10.1109/CVPR.2016.434
55. Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y.: Circle loss: A unified perspective of pair similarity optimization. IEEE Conference on Computer Vision and Pattern Recognition pp. 6397–6406 (2020), https://api.semanticscholar.org/CorpusID:211296865
56. Suomela, L., Kalliola, J., Edelman, H., Kämäräinen, J.K.: Placenav: Topological navigation through place recognition. In: IEEE International Conference on Robotics and Automation (2024)

57. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: InLoc: Indoor visual localization with dense matching and view synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 7199–7209 (2018)
58. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: InLoc: Indoor visual localization with dense matching and view synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
59. Taira, H., Rocco, I., Sedlar, J., Okutomi, M., Sivic, J., Pajdla, T., Sattler, T., Torii, A.: Is this the right place? geometric-semantic pose verification for indoor visual localization. In: IEEE International Conference on Computer Vision. pp. 4373–4383 (2019)
60. Tolias, G., Sicre, R., Jégou, H.: Particular Object Retrieval With Integral Max-Pooling of CNN Activations. In: International Conference on Learning Representations. pp. 1–12. International Conference on Learning Representations, San Juan, Puerto Rico (May 2016), https://inria.hal.science/hal-01842218
61. Torii, A., Arandjelović, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(2), 257–271 (2018)
62. Torii, A., Taira, H., Sivic, J., Pollefeys, M., Okutomi, M., Pajdla, T., Sattler, T.: Are large-scale 3D models really necessary for accurate visual localization? IEEE Transactions on Pattern Analysis and Machine Intelligence **43**, 814–829 (2021)
63. Torii, A., Taira, H., Sivic, J., Pollefeys, M., Okutomi, M., Pajdla, T., Sattler, T.: Are large-scale 3d models really necessary for accurate visual localization? IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(3), 814–829 (2019)
64. Trivigno, G., Berton, G., Aragon, J., Caputo, B., Masone, C.: Divide&classify: Fine-grained classification for city-wide visual geo-localization. In: IEEE International Conference on Computer Vision. pp. 11142–11152 (October 2023)
65. Vallone, A., Warburg, F., Hansen, H., Hauberg, S., Civera, J.: Danish airs and grounds: A dataset for aerial-to-street-level place recognition and localization. IEEE Robotics and Automation Letters **7**(4), 9207–9214 (2022). https://doi.org/10.1109/LRA.2022.3187491
66. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5022–5030 (2019)
67. Warburg, F., Hauberg, S., López-Antequera, M., Gargallo, P., Kuang, Y., Civera, J.: Mapillary street-level sequences: A dataset for lifelong place recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2623–2632 (2020). https://doi.org/10.1109/CVPR42600.2020.00270
68. Zeng, F., Jacobson, A., Smith, D.W., Boswell, N., Peynot, T., Milford, M.: Enhancing underground visual place recognition with shannon entropy saliency. In: IEEE International Conference on Robotics and Automation (2017), https://api.semanticscholar.org/CorpusID:196111988
69. Zhang, J., Cao, Y., Wu, Q.: Vector of locally and adaptively aggregated descriptors for image feature representation. Pattern Recognition **116**, 107952 (2021). https://doi.org/https://doi.org/10.1016/j.patcog.2021.107952, https://www.sciencedirect.com/science/article/pii/S0031320321001394
70. Zhang, Z., Sattler, T., Scaramuzza, D.: Reference pose generation for long-term visual localization via learned features and view synthesis. International Journal of Computer Vision **129**, 821–844 (2021)

71. Zhu, S., Yang, L., Chen, C., Shah, M., Shen, X., Wang, H.: R2former: Unified retrieval and reranking transformer for place recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 19370–19380 (June 2023)

MeshVPR: Citywide Visual Place Recognition Using 3D Meshes - Supplementary material

Gabriele Berton[1], Lorenz Junglas[2], Riccardo Zaccone[1], Thomas Pollok[3], Barbara Caputo[1], and Carlo Masone[1]

[1] Politecnico di Torino
[2] Karlsruhe Institute of Technology
[3] Fraunhofer IOSB

1 Examples of training and test images

In this Supplementary we present a large number of randomly chosen images from each of the training and test sets, shown in Fig. 1 (for the San Francisco datasets) and Fig. 2 (for the datasets from Berlin, Paris and Melbourne). These images allow to visually assess the large distribution gap between the various datasets, and, most importantly, between the synthetic and real images.

2 Further Qualitative Results and Failure Cases

Figure 3 shows examples of randomly chosen predictions on synthetic datasets for queries from Berlin, Paris and Melbourne, computed with the best model (SALAD + MeshVPR). In many cases it is visible how the two models (synthetic and real), given two images that are semantically similar but visually different, learn to map them nearby in the features space. For example the trees from the synthetic and real domains look very different, but when a query contains a tree, the predictions often contain a "synthetic" tree. A similar pattern is visible for statues (see examples from Paris), which are hardly recognizable in the synthetic domain, but are mapped in the same features space by the two models.

Fig. 1: Examples of images from the datasets from San Francisco, namely the real database, the High Quality (HQ) synthetic database, the Low Quality (LQ) synthetic database and the queries.

Fig. 2: Examples of synthetic database and real queries from the datasets of Berlin, Paris and Melbourne.

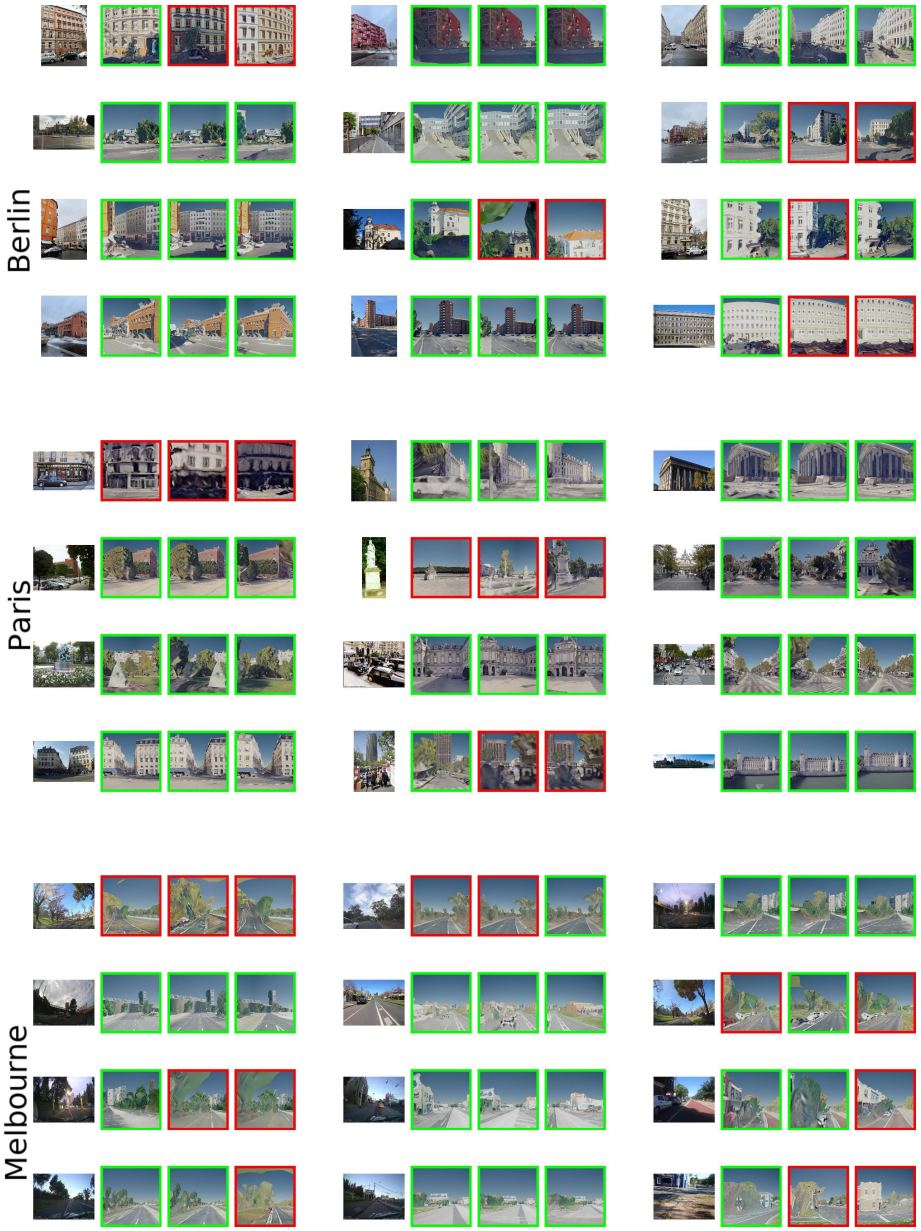

Fig. 3: Qualitative results from the three test sets, randomly picked, computed with the best model (SALAD + MeshVPR).