POLITECNICO DI TORINO

MASTER's Degree in
DATA SCIENCE AND ENGINEERING

MASTER's Degree Thesis

Understanding the needs of Image Retrieval for Visual Localization

Supervisors

Prof. Carlo MASONE

Dr. Gabriele Moreno BERTON

Dr. Gabriele TRIVIGNO

Candidate

Lorenzo SIBILLE

MARCH 2025

Understanding the needs of Image Retrieval for Visual Localization

Lorenzo Sibille

Abstract

Visual Localization is a key task in autonomous systems and robotics, consisting in the estimation of camera poses given their captures. Despite different approaches exist, they always rely on comparing the query image to reference images with known poses. To improve efficiency, relevant images to be matched are selected with retrieval pipelines. This work focuses on the impact of retrieval algorithms in Visual Localization. Firstly, current retrieval methods are benchmarked to determine the current state of the art on the task. Secondly, the needs of retrieval are investigated by selecting images using the known poses on different criteria. Lastly, different tasks related to Visual Localization are explored. The experiments highlight the limits of current approaches, as well as their margins of improvement and the future working directions to remove the retrieval performance bottleneck in Visual Localization.

ACKNOWLEDGMENTS

I would like to acknowledge and express my deepest appreciation to my supervisors Dr. Gabriele Moreno Berton, Prof. Mårten Björkman, Prof. Carlo Masone, Dr. Gabriele Trivigno, and my examiner Prof. Mats Nordahl. Without their expertise and precious guidance this thesis would not have been possible.

I would like to thank my family for their support, Alberto, Anna, Irene, Modesto, and Paola.

Lastly, I would like to thank everyone who made my degree journey special, in particular Andrea, Filippo, Gianluigi, Guido, Maddalena, Matteo, Maya, Moreno, Pietro, Stefano, and Tommaso.

Table of Contents

List of Figures

List of Tables

Acronyms

CNN	Convolutional Neural Network.
DELF	Deep Local Feature.
D2-Net	Detect-and-Describe Network.
LIFT	Learned Invariant Feature Transform.
R2D2	Repeatable and Reliable Detector and Descriptor.
SIFT	Scale Invariant Feature Transform.
SURF	Speeded Up Robust Features.
BoW	Bag-of-Words.
PCA	Principal Component Analysis.
VLAD	Vector of Locally Aggregated Descriptors.
GeM	Generalized Mean.
MAC	Maximum Activation of Convolution.
AP	Average Precision.
SALAD	Sinkhorn Algorithm for Locally Aggregated Descriptors.
PnP	Perspective-n-Points.
SfM	Structure-for-Motion.
RANSAC	Random Sample Consensus.

Chapter 1

Introduction

1.1 Visual Localization

Visual Localization is the task of estimating the position where an image was taken. The goal is to determine the position and the viewing angle of the capturing camera, resulting in six degrees of freedom which correspond to the so called camera *pose*. This a fundamental problem in augmented reality, robotics and autonomous systems, where accurate positional knowledge with respect to the surroundings is needed.

The Visual Localization pipeline adopted in this work is shown in Figure 1.1. The first step is to compute a reference system of coordinates and a set of spatial information. This is achieved by computing an offline 3D reconstruction given an unordered set of database images. The reconstruction is usually based on Structure-for-Motion (SfM) (Section 2.3), which matches 2D points between the images, and then given their geometric relations it is able to estimate their position in the 3D space as well as the poses of the cameras in a non-trivial optimization. If the database images are geo-tagged, the obtained reconstruction corresponds to the real world.

When a new image, referred to as the *query*, needs to be localized, the 3D points it sees are determined, and exploited with the corresponding 2D positions in the image plane to estimate the pose of the cameras within the reference system. While the procedure is conceptually straightforward, additional key concepts must be introduced, explaining how 2D points are computed and matched, and how the 3D points can efficiently be recovered.

Given an image, 2D keypoints are extracted and described. In particular, the goal is to find only relevant points, such as objects' corners, to reduce the problem dimensionality, to remove uninformative points, and to allow matching between interesting points reliably determined. Additionally, the extracted points have numerical descriptions, in order to be easily matched between different images comparing these representations. The obtained representation, including both coordinates and descriptors, is generally referred to as *local features* (Section 2.1).

The 3D points in the reconstruction contain also the local descriptors of the 2D points which originated them. Thus, initially, the correspondences between the 2D points in the query and the 3D scene were computed directly in the local feature

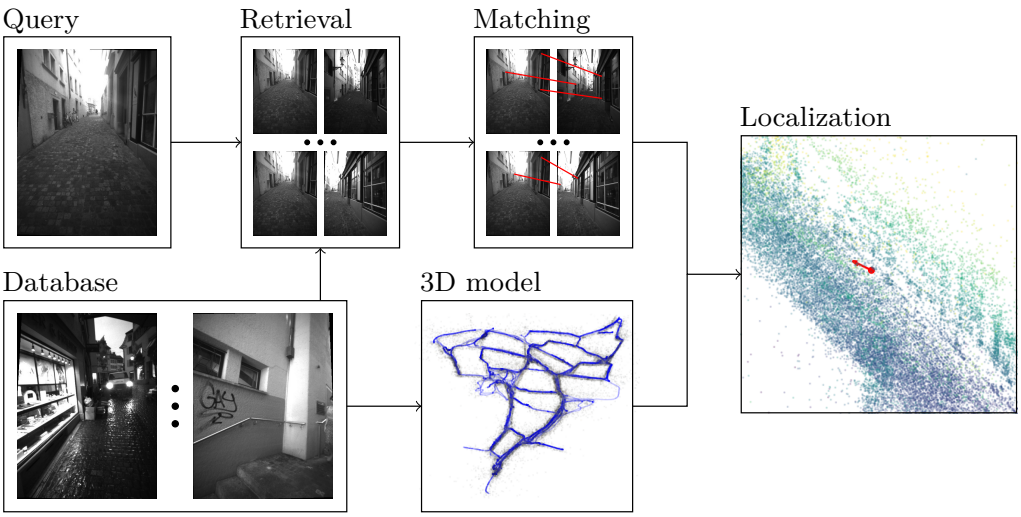

Figure 1.1: Hierarchical visual localization pipeline [3]. From the database, a global 3D model is built offline and stored. At query time, promising reference images are retrieved, then local matching is performed. The matched points are leveraged in 3D, and localization is achieved minimizing reprojection error in the query image.

space, comparing each 2D point to each 3D one. However, even though many works tried to improve scalability [1, 2], this procedure introduced a major performance issue. Currently, such direct approach has been substituted with a hierarchical [3] procedure, in which first the query is compared with database images, establishing 2D-2D correspondences, and then such correspondences are leveraged in the 3D space as the 2D points within the database have known 3D coordinates in the reconstruction. However, in the case of exhaustive matching between the query and all the database images, the complexity is not yet reduced. The key aspect to reduce computations is to reduce the number of database images tested, avoiding unnecessary computations for example on images not observing the same part of the scene.

To determine which database images are matched to the query, Image Retrieval techniques are used. Similarly to local features, whose descriptors are compared and points matched accordingly, the first step in a retrieval pipeline is to embed images into compact vectors, generally known as *global features* (Section 2.2). These vectors should represent the whole image, while highly reducing dimensionality with respect to the initial pixel space. The global features of the database are computed and stored offline, and at query time each of them is compared to the query representation, usually selecting the most promising images according to a metric distance on the embedding space.

To summarize the pipeline of Visual Localization, offline local and global features are extracted from the database images, and an SfM 3D model is built. When querying, local and global features are extracted from the query, then the Image Retrieval step determines, based on global features, which database images are promising to be matched; local features determine 2D-2D correspondences between query and database images, which are leveraged to 2D-3D correspondences using the reconstructed model, and pose estimation is computed on them.

1.2 Research question

While Image Retrieval is a necessary optimization, it has been shown in the past to limit localization performances [4]. Furthermore, Image Retrieval techniques are usually trained for Visual Place Recognition, which is the task to retrieve images containing the same landmark, without the need to show its same parts as no local matching is performed.

This work deeply analyzes the role of Image Retrieval in Visual Localization. In particular, the research questions investigated are:

- is current state-of-the-art retrieval limiting Visual Localization performances?
- which are the ideal retrieval characteristics that Visual Localization needs?
- do retrieval methods need to be tailored for Visual Localization?

1.3 Research methodology

To understand retrieval needs and to set upper bounds to be compared to retrieval methods, different criterion to select images exploiting the known query poses are defined (Section 3.2.1). For example, it is possible to select the database images closest to the query, enforcing some spread, or having a high visual overlap. Such methods replace the Image Retrieval step in Visual Localization, selecting database images according to the chosen criterion rather than the usual global features. While such approach is not applicable in real scenarios, as it requires the knowledge of target query poses, comparing their performances gives insights on the ideal needs that Image Retrieval should meet for Visual Localization; additionally, comparing such selection criteria to actual retrieval methods highlights the impact of Image Retrieval in Visual localization, possibly showing its limitation and that tailoring retrieval to the task is necessary.

Additionally, to understand how much rework of retrieval methods is necessary, a small pipeline variation is presented (Section 3.2.2). In particular, image retrieval techniques are adopted to select a higher number of database images with respect to the ones used in the actual localization, then they are further selected using the same criteria described above. Such procedure is compared to the standard retrievals, as well as to the ground truth based selection. This could show that small adjustments may be enough to remove the retrieval bottleneck from Visual Localization.

While there exist some benchmarks on Visual Localization [5, 6, 7], they are outdated in terms of dataset accuracy or retrieval algorithms. In this work, state-of-the-art and literature methods (Section 2.2) are extensively tested on accurate datasets (Section 3.1.4), evaluating and comparing the localization error of queries. Both quantitative and qualitative analysis of retrievals is performed, to give a human perspective on how much the gap with respect to the ground truth based upper bounds can be reduced, as they exploit labels unavailable in a real scenario and thus may not represent actually achievable results.

The experiments also test two additional tasks strictly related to the adopted Visual Localization pipeline (Section 3.2.3): Visual Localization using local maps, and 3D reconstruction. The first task is an alternative pipeline, which is needed when keeping a map built on the entire database is prohibitive. Such map is replaced by a smaller local map, built at query time, using only the retrieved images. The second task instead is the construction of the 3D map which is used in Visual Localization. While not being the main focus of this work, SfM methods need to get 2D-2D correspondences between reference images, and similarly to Visual Localization an exhaustive procedure is computationally unfeasible. Thus, to establish the point correspondences, each database image first goes through an Image Retrieval step, which determines to which other images it is matched. This part is tested on its own, evaluating the pose error of the images involved in the reconstruction, without further considering the impact of such reconstruction in the overall Visual Localization pipeline.

The contributions can be summarized as follows:

- the definition of different criteria to select database images exploiting ground truths, to understand retrieval needs and set performance upper bounds;
- the benchmark of above mentioned ground truth methods against actual image retrieval techniques, on Visual Localization and 3D reconstruction, to understand the current limitations of Image Retrieval in Visual Localization;
- the simple Image Retrieval variation consisting in retrieving more images and sampling them, to understand how much retrieval methods should be tailored for Visual Localization.

Chapter 2

Background

2.1 Local features

Local feature extraction is the process of detecting and describing relevant points of interest (*keypoints*) of an image. These local representations allow matching the same visual points within different images.

To be effective, the extracted keypoints should be *repeatable*, *i.e,* the same keypoint is found in images with different viewing conditions, *discriminative*, *i.e,* the representation is informative to unambiguously match keypoints, and *robust*, *i.e,* the representation should be invariant to changes in illumination and viewing condition.

While engineered approaches (Section 2.1.1) have been successfully adopted for years, they have recently been outperformed by learned representations based on Convolutional Neural Networks (CNNs) (Section 2.1.2).

2.1.1 Engineered features

Traditional approaches divide the task in detecting keypoints and describing them. Detection and description algorithms can thus be freely combined, depending on the application.

2.1.1.1 Detection

Detection algorithms extract meaningful keypoints of an image, which are traditionally associated to points in which the image is rapidly changing.

The Harris [8] operator performs corner detection finding points showing large but comparable gradient magnitude in two different directions. Blob detection algorithms further adopt second derivatives. For example, the extrema of the Laplacian operator or of the Determinant of the Hessian matrix can be adopted to locate keypoints.

These methods are computed in a scale-space representation, *i.e,* multiple times after applying successive Gaussian filters. The keypoints are then associated to the scale at which they are extracted, possibly needed in the description step. The Difference-of-Gaussians [9] detector directly finds keypoints while computing the

scale-space representation, approximating the Laplacian operator while being faster to compute.

2.1.1.2 Description

The most popular approaches are Scale Invariant Feature Transform (SIFT) [9] and Speeded Up Robust Features (SURF) [10].

SIFT [9] represent keypoints as histograms of the gradients computed in a local neighbourhood. The neighbourhood size is determined by the scale associated to the keypoint, achieving scale invariance; gradient directions are aligned to the strongest peak of the histogram, granting rotational invariance; histograms are normalized, increasing robustness to illumination changes. SIFT has been the most successful and adopted feature descriptor, inspiring many variations such as RootSIFT [11], which improves matching performances using normalized square root values of SIFT features.

SURF [10] adopts a similar strategy as SIFT, but the histogram is computed over fewer aggregated values of Haar wavelets rather than single gradients, allowing faster computations based on integral images. While being faster, the performances are slightly worse than SIFT.

2.1.1.3 Matching

Matching between local features is performed using Mutual Nearest Neighbors. Given two images, for each keypoint in an image the Nearest Neighbor in the other image is found, and viceversa. Generally, the Nearest Neighbor is the point giving the smallest Euclidean distance in feature space. If two keypoints are the Nearest Neighbor of each other, they are matched.

To enforce discriminative and unambiguous matching, usually the Distance Ratio between the Nearest Neighbor and the second Nearest Neighbor is thresholded.

2.1.2 Learned features

Learned feature extractors exploit the capability of Convolutional Neural Networks (CNNs) to efficiently compute dense maps, which are then refined and selected into sparse and informative keypoints. The first approaches, inspired by engineered methods, focused on single tasks. However, it quickly became clear that it was possible to exploit the same computations for the entire pipeline, jointly performing detection and description. In this way the efficiency was increased, and the bias on description caused by detection removed.

2.1.2.1 Detection and description

The first learned approach focusing on the end-to-end pipeline is Learned Invariant Feature Transform (LIFT) [12]. However, it still shows its engineered roots, computing detection, keypoint orientation and description sequentially. Starting from an image

patch, every step is computed on a sub-patch determined by the previous step, with a dedicated network.

Starting from Deep Local Feature (DELF) [13], a CNN dense map is extracted using a pre-trained backbone [14, 15, 16], and then it is further processed to perform both description and detection. In particular, DELF fine-tunes the dense map and performs detection with an attention layer. Although focusing on retrieval rather than local matching, DELF established the joint approach that all subsequent methods adopted.

SuperPoint [17] reduces the dimensionality with a CNN encoder, then one disjoint head for each task is trained, consisting in a small decoder followed by interpolation. Further details are discussed in Section 3.1.1.1.

Detect-and-Describe Network (D2-Net) [18] directly adopts the fine-tuned CNN dense map for detection and description, without any further processing. The map is seen as one feature vector per pixel, but also as one detection heatmap per output channel. A keypoint is selected if one of its value is a maximum for both its corresponding feature descriptor and detection map.

Similarly, Repeatable and Reliable Detector and Descriptor (R2D2) [19] adopts the dense map as descriptors, while it is further processed for detection. Two heatmaps are extracted with an additional layer, representing reliability and repeatability. Keypoints are extracted based and the product of those two scores.

All these approaches based on CNN dense maps handle scale invariance using an explicit image pyramid, *i.e,* keypoints and descriptors are computed at multiple scales, keeping the best ones overall.

2.1.2.2 Matching

Although the naive Mutual Nearest Neighbor (Section 2.1.1.3) is still a viable solution, learned matching techniques are the current state-of-the-art. For example, SuperGlue [20] exploits attention to iteratively contextualize descriptors, both within and across images, and then adopts these representations as cost values in the transport-like optimization problem to determine one-to-one correspondences. LightGlue [21] improves SuperGlue efficiency by early pruning unpromising points, performing an adaptive number of attention iterations, and matching based on linear scores rather than complex optimization problems. Further details are provided in Section 3.1.1.2.

2.2 Global features

Global features are compact representations that try to encapsulate characteristics of a whole image. Their compactness allows for quick image retrieval, computing similarity scores based on these short embeddings rather than the entire images. Although initially these representations were computed from local descriptors, now state-of-the-art approaches have dedicated CNN backbones which outputs are aggregated.

2.2.1 Aggregation of Local Features

A classic approach is to summarize multiple local descriptors into a single global representation.

For example, local features can be clustered into discrete visual words, and then aggregated into Bag-of-Words (BoW) representations which measure the frequency of each visual word in an image.

Other techniques which improve compactness and search time have been developed. A popular approach [22] is to exploit Fisher kernels [23, 24], an extended version of the BoW representation including higher order statistics, which dimensionality is heavily reduced using Principal Component Analysis (PCA). Similarly, Vector of Locally Aggregated Descriptors (VLAD) [25] starts from the BoW representation, in which each global descriptor contains, for each visual word, the sums of residuals of the local features that have such word as the nearest one. It can be seen as a simplification of Fisher vectors.

2.2.2 Aggregation of CNN dense outputs

CNN outputs are considered as dense local feature maps, which dimensionality is reduced used pooling. Such reductions are trainable, and different losses are defined depending on the model.

2.2.2.1 Pooling

NetVLAD [26] adopts a trainable generalized VLAD layer [25]. To achieve trainability, the layer must be differentiable, thus soft cluster assignment is exploited, and added as a weight in the sum of residuals to achieve global descriptors. As local features, it uses the dense output of a CNN backbone with dimensionality $W \times H \times D$ as $W \times H$ feature vectors D-dimensional. Further details are explained in Section 3.1.2.1. Similarly, the Sinkhorn Algorithm for Locally Aggregated Descriptors (SALAD) adopts the same approach, while solving the initialization bias, allowing to discard irrelevant local descriptors adopting a dustbin, and considering also cluster-to-feature properties casting the task as optimal transport. Further details are provided in Section 3.1.2.3.

Other approaches instead work on reducing channel feature maps into a single value. Given the output of dimensionality $W \times H \times D$, they deal with it as D feature maps with dimensionality $W \times H$, and the output of the pool layer will simply be D-dimensional. For each feature map, it is possible to extract the maximum value, as in Maximum Activation of Convolution (MAC) [27] and its multi-region form R-MAC [28], the mean, as in SPoC vectors [29] or a generalized mean with trainable parameters, as in Generalized Mean (GeM) pooling which is further detailed in Section 3.1.2.2.

2.2.2.2 Training losses

Many different losses are suitable options to train image embeddings for retrieval.

The triplet loss [30] trains on three images, namely a query image, a positive, *i.e,* relevant to the former one, and a negative example, with the goal to maximize the similarity between representations of the query and the positive example, while minimizing it for the negative case.

Similarly, contrastive loss [31, 32] trains on tuples. However, negative and positive samples are trained separately with different loss formulation, while having the same conceptual goal as the triplet loss.

Multi-similarity loss [33] considers many samples at the same time, both negative and positive. Furthermore, it is based on different similarities, both comparing the query with the database, as the previous losses, and comparing the database representations themselves, depending on their relevance.

Lastly, totally different approaches adopt a list-wise loss that consider many images at the same time, ranking them accordingly to a similarity score between their embeddings. Given the ground truth relevance to the query, and some expedients to keep such sorting differentiable [34], it is possible to directly train a model on the final retrieval metrics, such as Average Precision (AP). This kind of loss is further detailed in Section 3.1.2.2.

2.2.3 Retrieval

Depending on the chosen features, images are generally retrieved with Nearest Neighbor or Cosine Similarity [35]. Given a query image and a set of database images, the images retrieved have the smallest distance considering their global descriptors and the query one. Multiple distances can be adopted, such as Euclidean or Manhattan. Similarly, Cosine Similarity retrieves the images with the highest similarity, which is defined as the normalized inner product between two descriptors.

2.3 Structure from Motion

Structure-for-Motion (SfM) compute a 3D reconstruction from unordered sets of images. In the first stage, images are matched, and then using their 2D correspondences it is possible to determine their poses as well as the observed 3D points.

2.3.1 Correspondence search

Correspondence search determines scene overlap between images, trying to find projections of the same point in different images. This is achieved with the following steps:

- local features (Section 2.1) are extracted from each image;
- local features are matched between image pairs. Although the naive approach to consider all pairs would be possible, it is prohibitive for large collection and

thus images pairs are usually determined by a retrieval pipeline (Section 2.2) to determine promising image pairs;

- images with matching points are geometrically verified with homographies [36] or epipolar geometries [36]. Images are considered verified if enough inliers are found, according to some robust estimator such as Random Sample Consensus (RANSAC) [37].

2.3.2 Reconstruction

Reconstruction is handled differently between global and incremental approaches. Global methods, such as Theia [38], Open-MVG [39] and GLOMAP [40], estimate all the camera poses and observed points in a single large step, then performing a single bundle adjustment [41]. On the other hand, incremental approaches, such as Bundler [42], VisualSfM [43] and COLMAP [44], start from two-view reconstruction and gradually register new images, with multiple bundle adjustments during the process.

2.3.2.1 Incremental approaches

Given the scene graph obtained by the correspondence search, a reconstruction can be computed. Although each incremental method has its own characteristics, the procedure can be summed in the following steps:

- the model is initialized from two selected matching images, exploiting the estimated two-view geometry and triangulating the points;
- a new image is iteratively registered solving the Perspective-n-Points (PnP) problem [37], estimating the camera pose with respect to the reconstruction;
- the newly added image can extend the scene points, if it contains seen but unregistered points in registered images, via triangulation;
- every time the model registers a certain number of images, a bundle adjustment [41] is run, jointly optimizing camera poses and observed points.

COLMAP [44] is further detailed in Section 3.1.3.1.

2.3.2.2 Global approaches

The common pipeline for global approaches is as follow:

- from the view graph, camera poses are estimated at the same time. This is generally achieved by *rotation averaging*, aligning relative rotations into global absolute rotations, minimizing the error on the constraint $\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^T$, where $\mathbf{R}_i, \mathbf{R}_j$ are the absolute rotations and $\mathbf{R}_{ij}$ the relative one between images i and j. Then *translation averaging* is performed, factoring out the estimated absolute rotations from the poses, minimizing the error on the constraint $\mathbf{t}_{ij} = \frac{\mathbf{c}_i - \mathbf{c}_j}{||\mathbf{c}_i - \mathbf{c}_j||}$,

where $\mathbf{t}_{ij}$ is the relative translation and $\mathbf{c}_i, \mathbf{c}_j$ the absolute camera positions, which enforce coherent translation between images and the global coordinates.

- from two-view matches, given the estimated poses, a global triangulation is performed, estimating the positions of the observed 3d points.
- a global bundle adjustment [41] is run, jointly optimizing camera poses and positions of 3d points by minimizing the reprojection error.

Chapter 3

Method

3.1 Literature methods

In this section, the adopted methods existing in the literature are detailed. In particular, presenting the chosen local features (Section 3.1.1), global features (Section 3.1.2), SfM algorithm (Section 3.1.3), and the adopted datasets (Section 3.1.4).

3.1.1 Local features

Despite many valid local features exist, SuperPoint [17] has been chosen as it is still the state-of-the-art when combined with neural matching, since they are usually trained for it. Matching is performed with LightGlue [21], efficient and the state-of-the-art approach, pretrained for SuperPoint.

3.1.1.1 SuperPoint

SuperPoint [17] architecture, represented in Figure 3.1, reduces the image dimensionality through a CNN encoder, such as VGG [15]. In particular, given the original image size $H \times W$, the encoder computes F values over non overlapping 8×8 windows, obtaining a $H/8 \times W/8 \times F$ dimensional representation. SuperPoint then split the task of detection and description using two dedicated heads.

For detection, a decoder computes a $H/8 \times W/8 \times 65$ output, where 64 values are associated to each of the 8×8 pixels in a window, and the additional value is the dustbin which represent that the patch has no points; it allows for parameter-free upscaling to the initial resolution using sub-pixel convolution [45] after a softmax and removal of the dustbin. Description is achieved computing a semi-dense $H/8 \times W/8 \times D$ feature map, which is then upscaled to the full resolution using bi-cubic interpolation and L2-normalization.

The model is trained on synthetically warped images from MS-COCO [46], jointly training the detector with cross-entropy loss and the descriptor with a hinge loss.

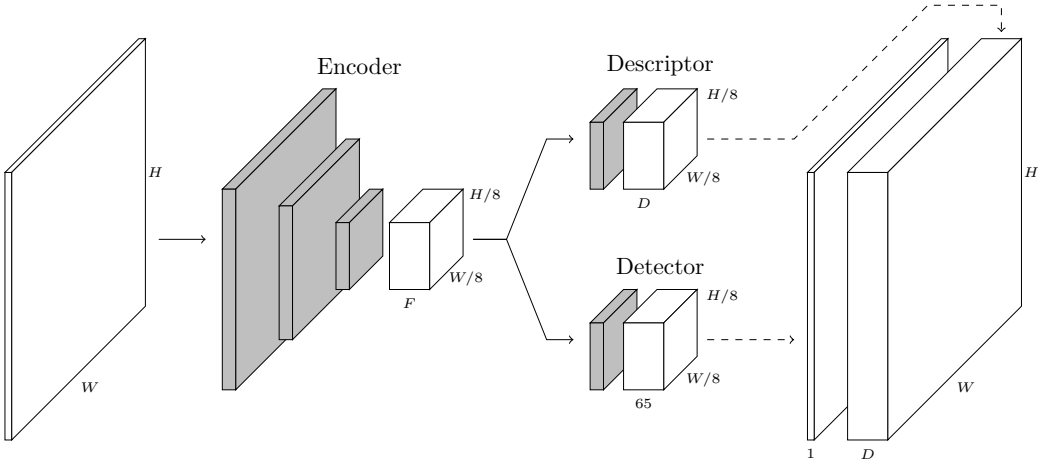

Figure 3.1: SuperPoint [17] architecture. Gray layers represent convolution, while dotted lines represent parameter-free upscaling, achieved with sub-pixel convolution [45] and bi-linear interpolation, respectively for detection and description.

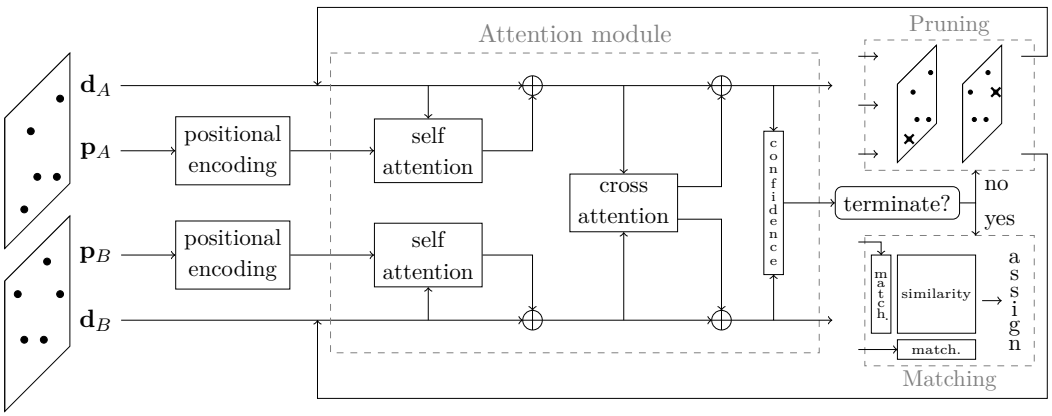

Figure 3.2: LightGlue [21] architecture. LightGlue achieves adaptability with a stack of identical attention modules. After each pass, termination is decided on estimated confidence for each point. If enough points are confident, then a fast matching based on pair-wise scores is performed. Otherwise, points confidently unmatchable are pruned, and the outputs are fed again in the attention module.

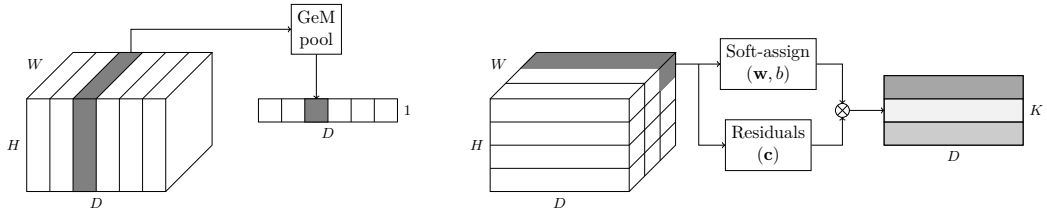

Figure 3.3: Comparison between AP-GeM [49], on the left, and NetVLAD [26], on the right. Both start from a $W \times H \times D$ dense map: AP-GeM deals with it as D $W \times H$-dimensional channel maps, and computes a single value on each of them via GeM pooling [51]; NetVLAD, instead, takes it as $W \times H$ D-dimensional local features, computing a VLAD [25] inspired representation based on residuals of learned visual words, weighted by a soft-assignment to mantain differentiability.

3.1.1.2 LightGlue

LightGlue [21] is able to adapt to the difficulty of images via an iterative approach. It is made of multiple identical modules, and after each stage a termination criterion is checked to avoid unnecessary computations for easy images. Each module, as shown in Figure 3.2, takes as input the coordinates of the points, with rotary positional encoding [47], and the descriptors, which are then updated for the following iteration using self-attention [48] followed by cross-attention, which contextualizes the embeddings with respect to the other points both in the same image and in the matching one.

Based on the embeddings, a compact fully connected layer computes a confidence value for each point, representing the reliability of the points, either in matchability or unmatchability. If enough points have a confidence higher than a certain threshold, the contextualization is ended and correspondences are computed. Otherwise, the points considered confidently unmatchable are pruned, and the updated embeddings are fed again into the module.

Matches are determined by two different scores: a similarity score is computed pairwise between all points in the two images, while a matchability score is assigned to each point base on its embedding. Each pair gets an assignment score proportional to the matchability of the points and the similarity score. All pairs with an assignment score greater than a threshold, and having the maximum similarity with respect to all pairs containing one of the two points, is selected.

3.1.2 Global features

The global features studied in this paper are NetVLAD [26] and AP-GeM [49], as they are the literature standards, and SALAD [50], which is the current state-of-the-art on Visual Localization.

3.1.2.1 NetVLAD

NetVLAD [26] is a trainable generalization of a VLAD layer [25]. VLAD descriptors aggregate a set of local features. As in the BoW approach, a set of K visual words $\mathcal{C} = \{\mathbf{c}_1, ..., \mathbf{c}_k\}$ is learned with k-means. Then, given an image with a set of N local

feature vectors $\mathcal{F} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, the global VLAD descriptor is a matrix $\mathbf{v}$ with dimension $K \times D$, where D is the dimensionality of the words in the codebook as well as the local features. Each element contains the sum of residuals of each local descriptor and the nearest word as

$$v_{i,j} = \sum_{a \in (1,N) | NN(\mathbf{x}_a) = \mathbf{c}_i} (x_{a,j} - c_{i,j}),$$

where $x_{i,j}$ represent the j-th element of $\mathbf{x}_i$, and the same notation is adopted throughout the entire section.

To achieve trainability, the NetVLAD pooling must be differentiable. Thus, a soft assignment to codewords is adopted, and the descriptor is adjusted accordingly as

$$v_{i,j} = \sum_{a=1}^{N} \frac{e^{-\alpha ||\mathbf{x}_a - \mathbf{c}_i||^2}}{\sum_{i'} e^{-\alpha ||\mathbf{x}_a - \mathbf{c}_{i'}||^2}} (x_{a,j} - c_{i,j}).$$

A further simplification is exploited, canceling the term $e^{-\alpha ||x_a||^2}$, obtaining

$$v_{i,j} = \sum_{a=1}^{N} \frac{e^{\mathbf{w}_i^T \mathbf{x}_a - b_i}}{\sum_{i'} e^{\mathbf{w}_{i'} \mathbf{x}_a + b_{i'}}} (x_{a,j} - c_{i,j}),$$

in which each cluster has a set of independent trainable parameters $\mathbf{w}_i, b_i, \mathbf{c}_i$.

To be invariant to the number of local features, both VLAD and NetVLAD representations are normalized, first using intra-normalization [52], $\mathbf{v}_i \leftarrow \mathbf{v}_i / ||\mathbf{v}_i||^2$, and then L2 normalization, $\mathbf{v} \leftarrow \mathbf{v} / ||\mathbf{v}||^2$.

NetVLAD takes as input a dense CNN map. In particular, given its shape $W \times H \times D$, it is considered as $W \times H$ D-dimensional local features. It was originally trained on AlexNet [14] and VGG-16 [15], with a triplet loss adapted to the weakly supervised scenario on images taken from Google Street View Time Machine [53].

3.1.2.2 AP-GeM

AP-GeM [49] combines GeM pooling with a listwise loss which directly optimizes the Average Precision (AP). GeM pooling [51] exploits the generalized mean [54] to extract one single value from each feature map. In particular, given a CNN output with shape $W \times H \times D$, it deals with it as D feature maps $\{\mathbf{x}_1, ..., \mathbf{x}_D\}$ with dimensions $W \times H$. The pooling results in a D-dimensional vector $\mathbf{f} = [f_1, ..., f_D]$, with

$$f_k = \left(\frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} x_{k,i,j}^{p_k} \right)^{\frac{1}{p_k}}.$$

Note that max pooling [27, 28] and average pooling [29] are special cases when $p_k \to \infty$ and $p_k = 1$, respectively. The parameter p_k is learnable, since this pooling is differentiable allowing for backpropagation.

The listwise loss considers the entire database at once. Given a query image I_q and a set of N database images $\{I_1, ..., I_N\}$, the similarity between the embeddings is the

vector S^q, where $S^q_i = sim(I_q, I_i)$. Additionally, the similarities can be sorted, and R denotes the ordered indexes such that the j-th image according to the similarity S^q is I_{R_j}. The ground truth image relevance is binary, and denoted as Y^q where Y^q_i is 1 if I_i is relevant to I_q and 0 otherwise. The Average Precision (AP) is then

$$AP = \sum_{k=1}^{N} P_k \Delta r_k,$$

where P_k is the precision at k, *i.e,* the fraction of relevant images in the first k sorted images according to R

$$P_k = \frac{1}{k} \sum_{i=1}^{k} Y^q_{R_i},$$

and Δr_k is the difference of recalls at k and at $k-1$, where the recall at j is the fraction of the relevant images in the first j and the total number of relevant images, *i.e,*

$$\Delta r_k = \frac{1}{\sum_{i=1}^{N} Y^q_i} Y^q_{R_k}.$$

The actual mathematical formulation is slightly relaxed to achieve differentiability, using soft assignment to bins rather than strict ranking [34]. The model was trained on the Landmarks dataset [30], using a ResNet-101 [16] pretrained on ImageNet [55] as backbone.

3.1.2.3 SALAD

Sinkhorn Algorithm for Locally Aggregated Descriptors (SALAD) [50] is heavily inspired by NetVLAD [26], while trying to solve some of its problems. The first main difference is the adoption of a visual transformer as the backbone, namely DinoV2 [56], which from each image extract n patches, resulting in the feature vectors $\mathbf{t}_1, ..., \mathbf{t}_{n+1}$, one per patch plus a global one. Then, SALAD directly addresses three problems of NetVLAD:

- The bias given by the k-means initialization in the soft-assignment is removed by assigning weight scores with two randomly initialized fully connected layers, in the form

  $$\mathbf{s}_i = \mathbf{w}_{s_2} \sigma(\mathbf{w}_{s_1} \mathbf{t}_i + \mathbf{b}_{s_1}) + \mathbf{b}_{s_2},$$

  where σ is a non-linear activation, and $\mathbf{s}_i$ an array containing the score with respect to each of the m clusters for the i-th patch;

- Some features might be irrelevant for the visual place recognition, while NetVLAD considers them all equally. SALAD introduces a dustbin to allow discarding features, by appending to each score a single learnable parameter z as $\bar{\mathbf{s}}_i = [\mathbf{s}_i, z]$;

- while NetVLAD only considers the feature-to-cluster relation, SALAD also models the cluster-to-feature one solving the assignment with the Sinkhorn

algorithm [57, 58], obtaining the final assignment matrix $\mathbf{P}$ after dropping the values corresponding to the dustbin.

The last differences is in an additional reduction of feature dimensionality as

$$\mathbf{f}_i = \mathbf{w}_{f_2}\sigma(\mathbf{w}_{f_1}\mathbf{t}_i + \mathbf{b}_{f_1}) + \mathbf{b}_{f_2},$$

obtaining the overall descriptor for the patches

$$v_{j,k} = \sum_{i=1}^{n} P_{i,j} \cdot f_{i,k}.$$

Lastly, $\mathbf{v}$ is flattened and concatenated with the reduced global descriptor

$$\mathbf{g} = \mathbf{w}_{g_2}\sigma(\mathbf{w}_{g_1}\mathbf{t}_{n+1} + \mathbf{b}_{g_1}) + \mathbf{b}_{g_2}.$$

The model was originally trained on GSV-cities [59], a large collection of scenes from Google Street View [53], using multi-similarity loss [33].

3.1.3 Mapping

COLMAP [44] has been the state-of-the-art SfM for the last 10 years, and all the existing code-bases are conveniently already adopting it. Additionally, the recent speed-up achieved by GLOMAP [40] is only on proper reconstructions, while triangulation and pose estimation are still handled by COLMAP.

3.1.3.1 COLMAP

COLMAP [44] follows the general incremental algorithm described in Section 2.3.2.1. However, COLMAP has some peculiarities which made it outperforming all the other SfM methods in the last decade.

- A multi-model geometric verification allows the optimal choice of the starting image pair, and to avoid degenerate triangulations between panoramic (*i.e,* with pure rotation) images. Both homographies and fundamental matrices are fitted, determining the number of inliers to each model; depending on their ratio, and possibly exploiting prior calibration, image pairs are classified as general, panoramic and planar, and the initial pair is chosen accordingly.

- The choice of the next camera to register within the partial model is optimized to maximize robustness. A common approach is to register the camera that sees the largest number of triangulated points; instead, COLMAP considers also the distribution of those points. Images are split into K bins in both dimensions, and then each bin containing at least one triangulated point contributes to a score with a weight of K^2. This operations is repeated L times, with the number of bins equal to $\{2^1, ..., 2^L\}$. The score keeps track of the distribution and the number of points, and the candidate image is chosen as its maximum.

- Transitive correspondences are exploited, merging the same matching point between multiple images into a single "feature track". Each feature track contains the coordinates of the point in the i-th image as well as the camera pose. Robustness is achieved using RANSAC [37], iteratively selecting a unique pair of images in the feature track, triangulating the point in the 3d space, and then computing in how many other images such 3d coordinates have a smaller reprojection error than a threshold t. Since a track may contain multiple independent points, the RANSAC process is run multiple times removing consensus points, allowing to split feature tracks.

- Bundle adjustment is run locally after each image, and globally every time the model grows of a certain percentage. Additionally, bundle adjustment is run iteratively in a pipeline which alternates bundle adjustment, filtering of observations with large reprojection errors, and re-triangulation. This improve the model robustness, removing outliers, and improving feature tracks certainty. Bundle adjustment is performed with Ceres solver [60].

- Since bundle adjustment is a major bottleneck in SfM, its optimization is crucial. In particular, cameras are clustered into small and highly overlapping groups. Each group is then parametrized as a single camera, highly reducing the number of parameters. However, images affected by the latest extensions are not clustered, allowing for stronger adjustments.

Although COLMAP has default feature extraction based on RootSIFT [11] and matching based on vocabulary trees, it allows to import matches based on external features and retrieval, which is crucial in this research. Additionally, COLMAP allows to perform triangulation, fixing known camera poses during bundle adjustment, rather than a proper SfM reconstruction.

3.1.4 Datasets

The dataset adopted are LaMAR [4] and VBR [61], which have extremely accurate known poses, fundamental to properly evaluate the impact of retrieval.

3.1.4.1 LaMAR

LaMAR [4] is a dataset for augmented reality containing images from different locations: HGE, the ground floor of a university building; CAB, a multi-floor office building; and LIN, few blocks of a small city center.

Each location contains many sequences, recorded at different times of the day and the year to increase diversity. The sequences have been collected with a Microsoft HoloLens 2 and an Apple iPad Pro, with additional custom sensors measuring depth and radio signals. The sequences are collected into different sets: a map set, two validation sets, and two test sets; while the map contains images from Hololens and iPad, the two devices have dedicated validation and test sets. Since the test sets

does not have public labels, and evaluation takes long time through LaMAR authors, all the experiments are conducted on the validation partitions.

Information from the different sensors is combined, refining camera poses up to the cm through a complex alignment within and across sequences. Such accuracy, combined with the scale, density, and diversity, makes LaMAR a perfect set to experiment Visual Localization.

3.1.4.2 VBR

VBR [61] is a dataset containing different locations in Rome. The dataset is developed for SLAM, and thus it has single but long sequence per location. However, there is a big variety in the locations themselves, having interiors, exteriors in urban areas and exteriors in green areas.

The dataset has two different cameras, an IMU, a LiDAR, and a RTK-GPS, which, as for LaMAR, allow for accurate poses through refinement.

Since the dataset has single sequences per scene, the sequences are split into reference and test images, as explained in Section A.1.

3.2 Contribution

In this section the contributions are described. In particular, the definition of retrievals based on ground truths (Section 3.2.1), the pipeline variation combining their sampling criteria with existing retrieval methods (Section 3.2.2), and a brief overview of the experiments (Section 3.2.3).

3.2.1 Ground truth based retrieval

To understand the needs of retrieval in Visual Localization, methods selecting images based on different criterion, exploiting ground truths, are tested. These methods replace the usual Image Retrieval stage in the Visual Localization pipeline, deciding which database images should be matched given a query.

While some of them do not have any practical application, as they exploit also the target query pose, they are extremely important to understand the objective actual retrieval methods should have. This information can lead, for example, to new training procedures tailored for Visual Localization. Furthermore, they may highlight the limits of actual Image retrieval methods, setting loose upper bounds on localization performances.

While the goal is to select images from the entire database, some methods are computationally intensive and thus an initial reduction of the number of database images to select from is needed. The proposed methods first start from *candidate* images, that ideally could be the entire database but here it is a subset (Section 3.2.1.1), and then the best candidates according to some criterion (Section 3.2.1.2) are sampled, and used in the Visual Localization pipeline.

3.2.1.1 Candidates

As some of the proposed selection criteria are computationally expensive, candidates are initially reduced. This is necessary to limit the computational resources needed, while maintaining exhaustivity. Furthermore, the candidates are the same for all the samplings, and thus determined only once and stored to be used in multiple runs.

For each query image, the candidates are the database images within 20 meters from the query, with a maximum difference in the viewing angle of 120°, and with frustum intersection, *i.e,* the intersection between the fields of view, within 20 meters. Given the nature of the datasets, extremely dense and depicting relatively closed spaces, these parameters are suitable to reduce computations without loosing generality. Additionally, only database images observing at least 5% points of the query image are considered. Again, generality is not lost since images with a smaller number of seen points are unlikely to contribute to localization, while many outliers which slow selection are removed.

3.2.1.2 Selection

Different methods to sample from candidates are defined:

- *random*: candidates are randomly sampled. This aims to distinguish between the impact of sampling using specific criteria, and the impact of the initial definition of candidates on its own. To get a fair comparison, and since this method is extremely sensitive to outliers, isolated database images are removed from candidates. Isolated images are defined through DBSCAN-like [62] clustering on candidates' poses, and then clusters containing few cameras are discarded.

- *pose-near*: the images closest to the ground truth position of the query are selected. At this stage, only the translation are considered, as the intersection of field of view is already verified in the definition of the candidates.

- *pose-coverage*: the images are taken such that they are spread in space as much as possible. The first image selected is the closest to the query pose, and then iteratively the image having the maximum distance from the nearest already selected image is taken. Formally, given the set of candidate images $\mathcal{C}$, the set of already selected images $\mathcal{R}$, and the position of the i-th image as $\mathbf{x}_i$, at each step the selection is

$$\mathcal{R} \leftarrow \mathcal{R} \cup \left\{ \max_{i \in \mathcal{C} \setminus \mathcal{R}} \left\{ \min_{j \in \mathcal{R}} \{ |\mathbf{x}_i - \mathbf{x}_j| \} \right\} \right\}.$$

  Since this method is sensitive to outliers, the same removal as for *random* is adopted. Furthermore, in some experiments, an additional restriction of the candidates being within t meters is used, obtaining an hybrid with *pose-near*, trying to force spread but closer to the query.

- *covisibility*: the images observing the highest number of points appearing in the query image are selected.

- *covisibility-coverage*: the images are selected such that their corresponding points cover as much as possible the points appearing in the query. Iteratively, to each image is assigned a score s, and then the image with the maximum score is selected. The score is the sum, for each point matching the query, of the inverse of how many times such point already appears in the sampled images. Formally, given the points $\mathcal{P} = \{p_1, ..., p_n\}$ appearing in the query, the set of already selected images as $\mathcal{R}$, the set of candidate images $\mathcal{C}$, δ_{ji} is defined such that $\delta_{ji} = 1$ if p_j appears in the i-th image and 0 otherwise. Then, the score of the i-th image is computed as

$$s_i = \sum_{p_j \in \mathcal{P}} \frac{1}{1 + \sum_{l \in \mathcal{R}} \delta_{jl}} \delta_{ji}.$$

  This score is proportional to how many matching points a candidate image has, while also giving more importance to points not yet retrieved.

These methods will be deeply tested in Section 4, replacing the usual Image retrieval step, to understand which ideal criterion is more relevant and beneficial in Visual Localization. In all the experiments, the further speed-up of considering for all samplings only the 50 candidates with the highest covisibility has been adopted. Partial test were run without such assumption, without observing major differences while being drastically slower.

3.2.2 Sampling retrievals

While pose based criteria described in Section 3.2.1.2, namely *pose-near* and *pose-coverage*, heavily depends on target ground truths and thus meaningless in real applications, covisibility methods, namely *covisibility* and *covisibility-coverage*, only depend on labels in the initial definition of candidates. Thus, such methods are usable also in real applications if candidates are properly defined without exploiting any a-priori knowledge.

In this work, such sampling criteria are tested on candidates determined by actual retrieval. For example, it is possible to retrieve N images with SALAD [50], and then sampling $n < N$ images to match. This is useful to determine if a complete rework of retrieval methods is necessary, or if small pipeline adjustments or fine tuning may be sufficient.

Unfortunately, covisibility introduces a big computational overhead even in case of parallelizable neural matching as LightGlue [21]. However, if future works may find faster effective sampling, the overall process may be more efficient than keeping the whole N images, balancing the sampling computations with faster localization, as it scales with the number of matched images.

3.2.3 Benchmark

Although there exist some benchmarks for Visual Localization [4, 5, 6], they are outdated in terms of retrieval methods or using inaccurate datasets. This work, on the other hand, benchmarks the two literature standards AP-GeM [49] and NetVLAD [26] against the state-of-the-art retrieval SALAD [50], on datasets accurate to the centimeter [61, 4]. Additionally, a thorough study of the retrieval needs is performed exploiting ground-truth based retrievals explained in Section 3.2.1, setting loose upper bounds on the performances. Lastly, small pipeline variation based on sampling from retrievals (Section 3.2.2) are tested, showing if major reworks are needed or small adjustments are enough to reduce the gaps with respect to the computed bounds.

The main task analyzed is Visual Localization from a global 3D map, in which localization is performed as described in Section 1.1. However, inspired by [7, 6], two additional tasks are explored: Visual Localization based on local maps, in which a local 3D reconstruction is built at runtime only on the retrieved images, and 3D reconstruction itself, in which COLMAP [44] is run without exploiting known poses, and the model is then rigidly aligned to the ground-truths using RANSAC [37]. These additional tasks are strictly related to Visual Localization on global maps, and thus their retrieval needs are compared.

Chapter 4

Experiments

4.1 Visual Localization

In this section, Visual Localization is performed using global maps, as described in Section 1.1. Given reference images, a global 3D reconstruction is triangulated with COLMAP [44], using local SuperPoint features [17] matched with LightGlue [21]. For each image, the 50 most promising images according to NetVLAD [26] are matched to build the view graph. At query time, N reference images are retrieved from the database with the chosen method, matching SuperPoint features with LightGlue as for the mapping. Then, based on the matched points leveraged in the 3-dimensional space, queries can be localized within the global reconstruction.

4.1.1 Quantitative analysis

The first experiments are on the LaMAR [4] dataset, on all the validation sets, testing NetVLAD [26], AP-GeM [49], and SALAD [50], as well as the ground-truth based methods described in Section 3.2.1.

Different numbers Ns of retrieved images have been tested, as shown in Figure 4.1. Increasing N improves localization performances, at least for actual retrievals: in fact, retrieving more images, there is a higher chance of finding positives, which enhance localization; at the same time, however, there is a higher chance of finding irrelevant images, but their noise is mitigated using robust estimators such as RANSAC [37]. At the same time, for ground-truth bounds, the impact of N is smaller, meaning that ideally localization could be performed with a small value of N. This is important as localization complexity scales with the number of matched points, highly correlated with the number of retrieved images. Until saturated performances comparable to the ground-truth based retrieval are achieved, a trade-off between accuracy and pose estimation time is unavoidable. Additional results varying N are reported in Tables A.2, A.3, and A.4.

In Figure 4.2 the localization for $N = 10$ results on LaMAR HGE are reported, while complete results on the entire dataset are reported in Figures A.2 and A.3. Although specific results depend on the scene, overall SALAD significantly outperforms NetVLAD and AP-GeM, which yield similar performances. However, there is still

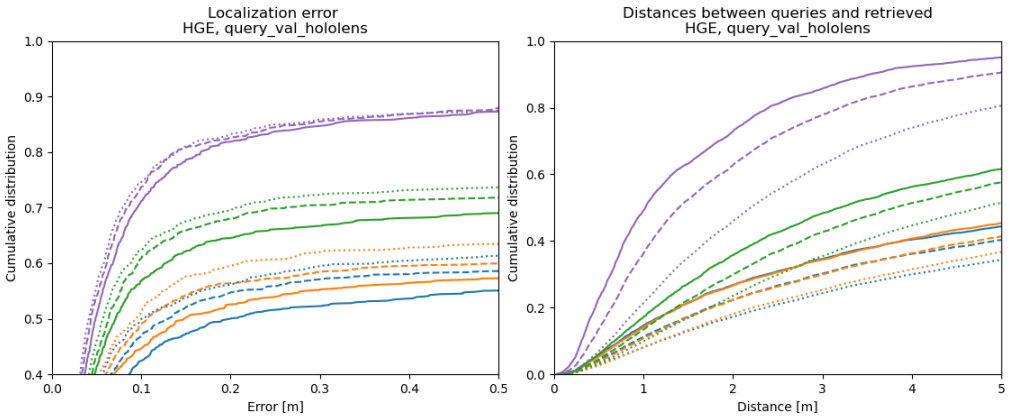

Figure 4.1: Localization error varying the number of retrieved images in LaMAR [4] HGE. The comparison is between NetVLAD [26] (■), AP-GeM [49] (■), SALAD [50] (■), and the ground-truth based (Section 3.2.1.2) *pose-near* (■). Solid lines represent the results obtained retrieving 5 images, while dotted and dashed lines retrieving 10 and 20 images, respectively.

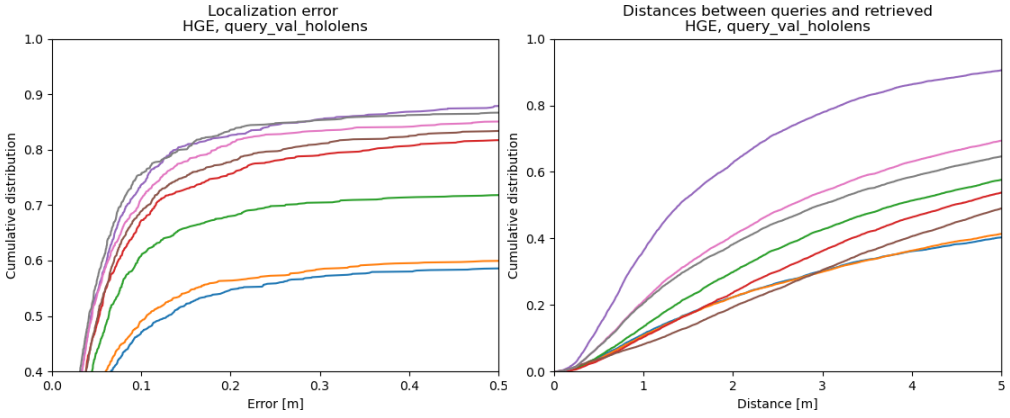

Figure 4.2: Localization error and distance between queries and retrievals for Hololens validation images in LaMAR [4] HGE scene. The comparison is between actual methods, namely NetVLAD [26] (■), AP-GeM [49] (■), SALAD [50] (■), and retrievals exploiting the ground-truths (Section 3.2.1.2): *random* (■), *pose-near* (■), *pose-coverage* (■), *covisibility* (■), *covisibility-coverage* (■). The results are obtained retrieving 10 images.

a big performance gap with respect to ground-truth based retrievals. Notably, all of them have comparable performances, even *random* selection, which means that the process to establish candidates (Section 3.2.1.1) is valid and high covisibility is a fundamental need in retrieval.

Instead, there is no major benefit from the positional spread of retrieval as enforced by *pose-coverage* sampling, while *pose-near* and *covisibility-coverage* samplings generally perform slightly better than the other. A part from the results on the iPad images for the CAB scene, in which some repetitive elements (corridors, stairs, classrooms) corrupt covisibility and make *pose-near* stand out by finding the correct elements within repetitions, there is no proper correlation between performances and closeness of query and retrieved images. In fact, *pose-near* has the closest retrieval by a big margin, but the performances are comparable to the other ground-truth retrievals. Additionally, *pose-coverage* and *random* have retrieval distances higher than SALAD, while outperforming it. The performances of *pose-near*, considering that the candidates are already selected for covisibility, are explained by the intrinsic correlation between closeness and similarity of viewpoints, which means that many points are covisibile. Furthermore, since *covisibility* and *covisibility-coverage* do not exploit the ground truth, a part from the proposal of candidates in which it is exploited only as a speed-up solution, they have higher potential application in a real case scenario, such as described in Section 3.2.2.

The same experiments have been run on VBR [61], as shown in Figure A.4. Unfortunately, the density and the scenes registered in single sequences make the retrieval task trivial: with few exceptions, namely SALAD in *spagna*, and the *colosseo* scene, results are almost independent from retrievals and variations are spurious. Within ground-truth methods, *covisibility-coverage* and *pose-near* are the best performing, although by a tiny margin.

4.1.2 Qualitative analysis

To understand if retrieval improvements seem possible from human perspective, a qualitative analysis is performed. The main challenges are the lack of information in an image and the ambiguity due, for example, to repetitive structures, as exampled in Figure 4.3. While uninformativeness is generally hard in any retrieval context, ambiguity is a problem more specific to Visual Localization, since retrieving the same object is not enough, needing the same specific section to allow matching.

Despite these two challenges, many negatives, *i.e,* badly localized queries, have room for improvement in the retrieval stage, from a human standpoint, having some details removing ambiguity or uninformativeness and allowing for correct retrieval, such as paintings, signs, or geometric relations. Some examples are shown in Figure A.5, and analyzed more in detail in Section A.3.

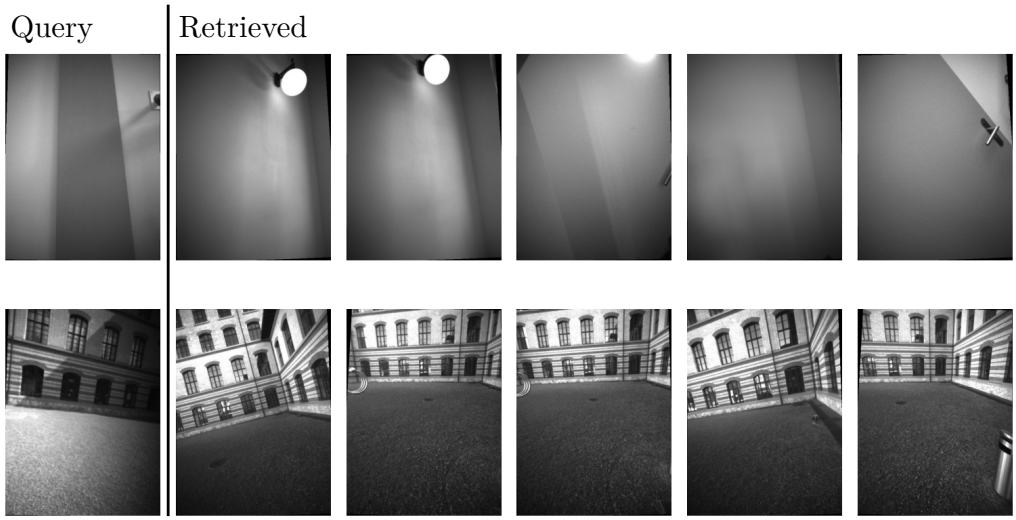

Figure 4.3: Examples illustrating the main challenges in retrieval: uninformativeness and ambiguity. The retrieval shown is SALAD [50], on the CAB scene of LaMAR [4]. More examples are shown in Figure A.5.

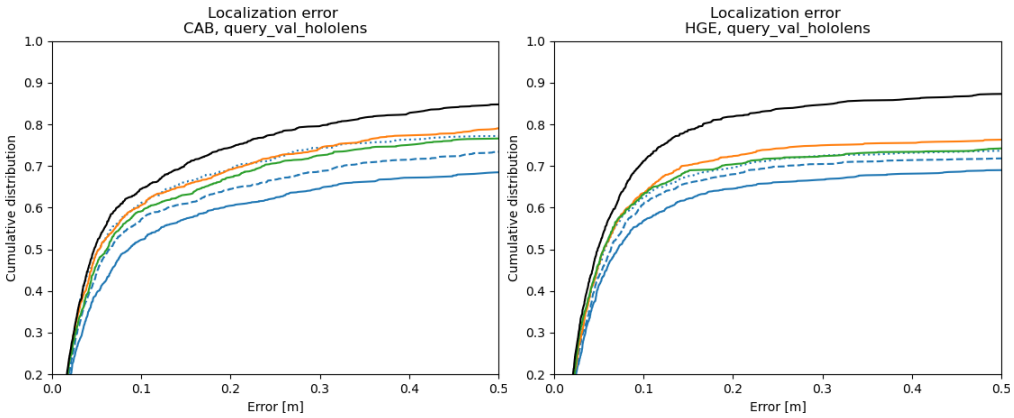

Figure 4.4: Localization error in LaMAR [4], on HoloLens images in scenes CAB an HGE. The comparison is between plain SALAD [50] (■) with 5, 10, and 20 retrieved images (respectively, solid, dashed and dotted lines), the ground-truth baseline obtained selecting 5 images with *pose-near* (Section 3.2.1) (■), and sampling 5 images among the 20 retrieved with SALAD using *pose-near* (■) and *covisibility* (■). Results on the other LaMAR scenes are shown in Figure A.6.

4.1.3 Sampling retrievals

It is possible to adopt the samplings described in Section 3.2.1 on candidates based on actual retrieval methods. For example, in Figure 4.4, 20 images are retrieved with SALAD [50], and then only 5 images are selected on *pose-near* and *covisibility* criterion. Additional results are shown in Figure A.6.

Sampling 5 images in this way improves performances not only with respect to directly taking the best 5 images according to SALAD, but also 10, and obtaining results comparable to using the whole 20 images in the localization phase. Despite the ground-truth upper bound is still far, such result shows that the key is not the number of images retrieved, but rather their quality, and at the same time that the current retrieval methods may have major improvements with tiny modifications, such as fine tuning. Additionally, sampled localizations are sometimes better than using all the images, showing that increasing the number of retrieved images does not always correspond to an improvement, introducing outliers, despite robust RANSAC [37] pose estimation.

The proposed samplings are not suitable for real applications, since *pose-near* exploits the targets, and *covisibility* introduces the overhead of matching all the 20 images, which is parallelizable but still computationally intensive. However, a similar approach may be exploited if efficient and effective sampling criterion are found, as localization complexity benefits from fewer matched images.

4.2 Visual Localization with local maps

As proposed by [7], it is possible to perform Visual Localization on small local maps instead of keeping huge global maps. This approach, while increasing computations at query time by building the local maps, is necessary when the scale makes a global map prohibitive. Local maps are built using only the images retrieved for the query, triangulating the points exploiting the known poses.

Results are reported in Figure 4.5. In this section, the candidates for *pose-coverage* are first filtered to be within a certain adaptive radius from the query, starting from 3m and iteratively increasing it by 50% until the candidates kept were at least twice the number to be sampled, in order to not degenerate *pose-coverage* into *pose-near*. This was necessary due to the sensibility of noise in *pose-coverage*, which in this task corresponds to poor triangulations impacting localization.

This approach on local maps has a lower recall of 5-10% on average with respect to using global maps, depending on the retrieval and the dataset. Using more reference images may improve the quality of the local maps, boosting performances, but at the same time increasing triangulation time and matching time between database images, if not done offline. At the same time, for some applications it may be the only suitable choice, and the performance drop can be acceptable at large scales in which accuracy at the decimeter is not necessary.

Unexpectedly, having a wider distribution of reference images is worse than

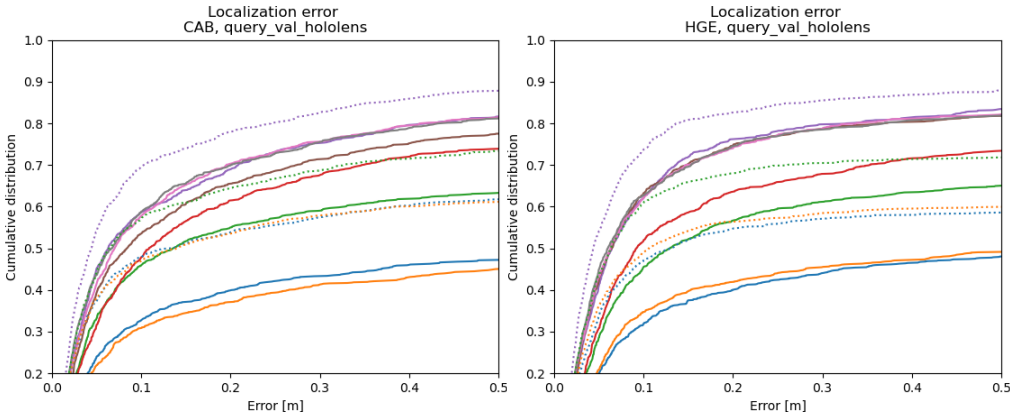

Figure 4.5: Localization error for validation images in LaMAR [4] CAB and HGE scenes, using local maps. The comparison is between actual methods, namely NetVLAD [26] (■), AP-GeM [49] (■), SALAD [50] (■), and retrievals exploiting the ground-truths (Section 3.2.1.2): *random* (■), *pose-near* (■), *pose-coverage* (■), *covisibility* (■), *covisibility-coverage* (■). For *pose-coverage* the images are filtered to be within an adaptive radius from the query, starting from 3m and increased by 50% iteratively if not enough images are kept. Dotted lines are obtained using global maps, to ease comparison between the two approaches. The results are obtained retrieving 10 images. The other LaMAR scenes are reported in Figure A.7.

retrieving the closest to the query. Even the modified *pose-coverage* approach previously described, having a close spread, performs worse than *pose-near*. This is probably due to the exploitation of ground truth in building the map, performing only triangulation rather than a reconstruction, which would have a badly conditioned pose estimation step for very close images. In this scenario, *pose-near* and covisibility based methods have similar performances. Overall, the needs of retrieval seem to be the same for Visual Localization using global maps and local maps: high covisibility, which is generally correlated to closeness, if as in this work selected images are filtered granting a minimum visual overlap.

4.3 3D Reconstruction

The quality of 3D reconstruction is strictly related to Visual Localization, especially when the poses of reference images are unavailable. In this section, the same retrievals benchmarked for Visual Localization are tested in 3D reconstruction. The experiments are run only on LaMAR [4] Hololens images, since iPad images are colinear and thus reconstruction is ill-posed. Reconstructions are achieved using COLMAP [44], and the obtained reconstruction is rigidly aligned to the ground-truth poses using RANSAC [37]. Being computationally intensive, requiring days for each reconstruction on more than 10000 of images, the scenes analyzed are only the smaller validation sets rather than the actual reference images. As evaluation metric, only the error in the poses is adopted. Other measures, such as the reprojection error, depend on the number of points triangulated, making a fair objective analysis impossible.

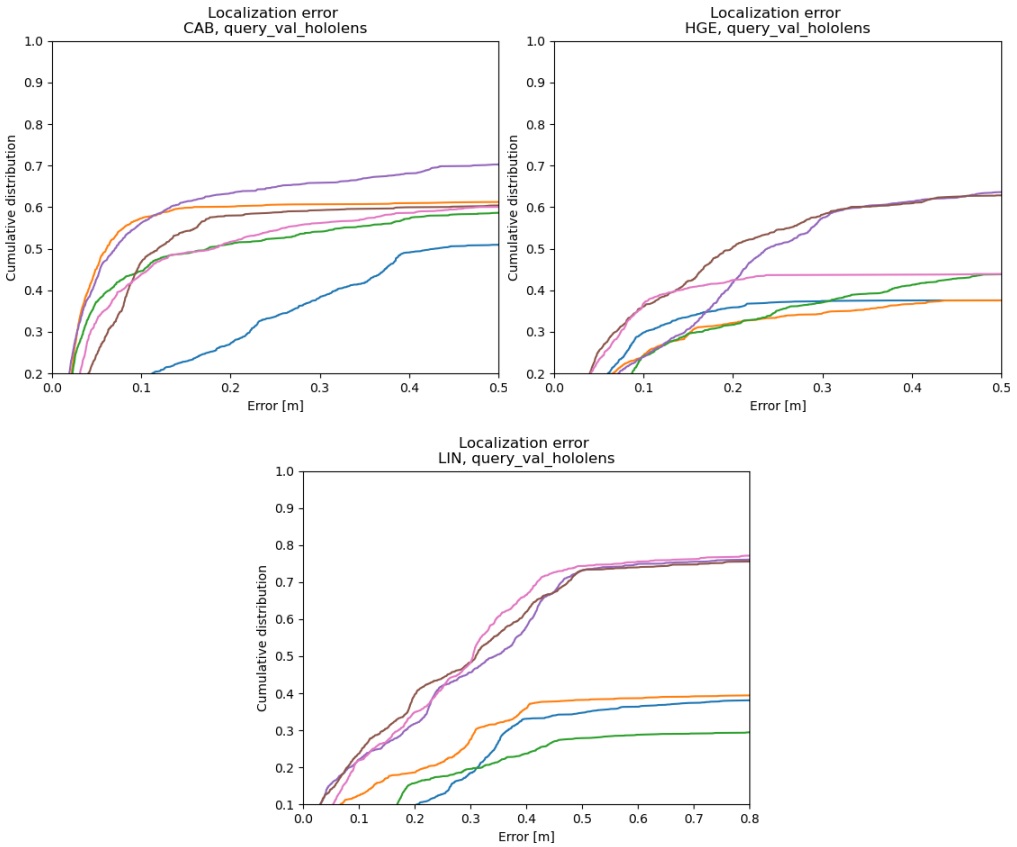

Figure 4.6: Reconstruction pose error for Hololens validation images in LaMAR [4]. The comparison is between actual methods, namely NetVLAD [26] (■), AP-GeM [49] (■), SALAD [50] (■), and retrievals exploiting the ground-truths (Section 3.2.1.2): *pose-near* (■), *pose-coverage* (■), *covisibility* (■). For *pose-coverage* the images are filtered to be within an adaptive radius from the query, starting from 3m and increased by 50% iteratively if not enough images are kept. The results are obtained retrieving 10 images.

Results are shown in Figure 4.6. Despite validation sets containing fewer sequences, making the retrieval easier, there is still a big performance gap with respect to ground-truth based upper bounds, depending on the scene. Notably, in this task, there is no major difference between NetVLAD [26], AP-GeM [49], and SALAD [50]. Instead, as the previous experiments, the most consistent ground-truth approach is *pose-near*, highlighting once again that the need is to retrieve close images. Although intuitively in reconstruction a higher pose diversity should allow for more robustness, COLMAP exploits the view graph to transitively match points between image pairs: thus, the good performances of *pose-near* are probably tied to the creation of a highly connected view graph. Instead, the struggles of *covisibility* are probably due to image pairs with high covisibility being co-linear, creating poorly connected graphs given split between front and side cameras.

Chapter 5

Conclusions

5.1 Conclusions

The current retrieval methods are limiting Visual Localization performances. Despite the ground-truth upper bounds are just ideal, some of them only exploit the known poses in the candidate proposal, showing that there is margin of improvements only based on appearances, as also highlighted in the qualitative analysis, despite facing challenging ambiguity and uninformativeness. In particular, the main need of retrieval for Visual Localization, according to the experiments, is finding images with high covisibility, possibly covering different parts of the query; this is often granted finding very close images facing the same direction.

This shows the potential path to improve retrieval for Visual Localization. In fact, current Image Retrieval methods are trained on Visual Place Recognition, which needs invariance with respect to the viewpoint, while the experiments showed the benefits of viewpoint similarity. A possible working direction could be to fine tune existing methods trying to consider the camera position with respect to the observed scene in the embeddings, and possibly to focus also on details that are usually lost in producing global features, as they are crucial for ambiguous query. Additionally, the experiments show the benefit of further selecting images after retrieval, at least on current methods. Although an efficient sampling strategy is crucial for the task, localization speed would improve with respect to using all the images, while outperforming a traditional retrieval with an equal number of images. Using a small number of images is sufficient, as performances saturates quickly when increasing it, especially in case of good retrievals.

While classic literature standards are still used in recent works, given the performances on Visual Localization new studies should focus on less traditional models in favor of actual state-of-the-art approaches, even if more complex.

Reconstruction, on the other hand, showed different retrieval needs, more related to pose similarity than covisibility. However, it is believed that this is due to the nature of the dataset, and further studies are necessary. Additionally, reconstruction seemed more invariant to retrieval, probably due to the transitive matching abilities of COLMAP [44].

5.2 Limitations

The datasets adopted in this work are sequential dataset. The study needs extremely accurate reference poses on a large scale, and unfortunately these two characteristics are not easy to find on non-sequential data, as large scale labels are often computed with SfM compromising their accuracy and affecting their analysis. However, the adoption of sequential data makes this study less generalizable on other kind of data.

As the focus of this study was the retrieval stage, the other aspects have been fixed. In particular, local features and matching have been chosen according to the state-of-the-art, but may have still affected evaluation. Furthermore, all the experiments on Visual Localization have been performed fixing the reference map built using Netvlad [26]; while the scale of the datasets, the transitive abilities of COLMAP [44], the exploitation of known poses, and the high number of retrieved images should make the results almost invariant to such choice, it still may have introduced a bias.

In this work covisibility has been defined according to matched local features, as it is the most immediate approach. However, other definitions [4] exploiting the reference poses exist and are possibly more reliable, not being influenced by repetitive patterns and false matches. While possibly not adoptable in real applications, they may still provide insights and drive innovations.

Lastly, the study of 3D reconstruction has been carried with COLMAP, which does not allow for large scale scenarios due to complexity and repeated bundle adjustments, and thus smaller subsets have been adopted. However, GLOMAP [40] should allow for larger scales in reasonable time complexity, although at the time of this work it had compatibility issues due to early development. Larger scale studies may provide results impossible to see in the smaller subsets tested in this work.

5.3 Future Work

Future works should focus on extending the study according to the limitations (Section 5.2). In particular, large scale non-sequential datasets should be adopted, while testing different local features and matching, different maps in Visual Localization, different covisibility definitions, and different and more efficient methods in the 3D reconstruction. Such extension may provide additional insights, further validating the conclusions drawn in this work or, possibly, disconfirming them.

If the same conclusion are drawn from extended studies, then the focus should be on tailoring retrieval models for Visual Localization. Given the conclusions in this work, the suggested approaches are fine tuning existing retrievals, or slight pipeline variations as the additional sampling step proposed in this study. Such research may definitely remove the bottleneck of retrieval performances in Visual Localization, allowing to refine other steps of the pipeline.

Appendix

A.1 VBR splits

Since VBR [61] contains only one labeled sequence per scene, a split into training and validation images is needed. Fortunately, these long sequences pass multiple times through the same places, allowing to split into sub-sequences with spatial overlap. Examples of adopted splits are shown in Figure A.1. Given a sequence, the test set is made of images captured by the left camera within a sub-sequence from a starting timestamp to an ending timestamp, as reported in Table A.1. For *ciampino* scene, two subsequences are used as the test set. The reference set, used to build the 3D models, consists of all the timestamps outside the test, using both left and right cameras. Test sequences are reduced to 750 images, uniformly sampled within the time dimension.

A.2 Quantitative analysis of retrieval and localization

Figures A.2 and A.3 illustrate the localization error and the distances between queries and retrieved images, considering only translation distance and not rotation. While NetVLAD [26] and AP-GeM [49] have similar results, they are outperformed by SALAD [50]. SALAD, on the other hand, is able to reduce the gap with the ground-truths based loose upper bounds, even achieving comparable performances depending on the dataset. CAB is a very repetitive dataset, containing ambiguous buildings,

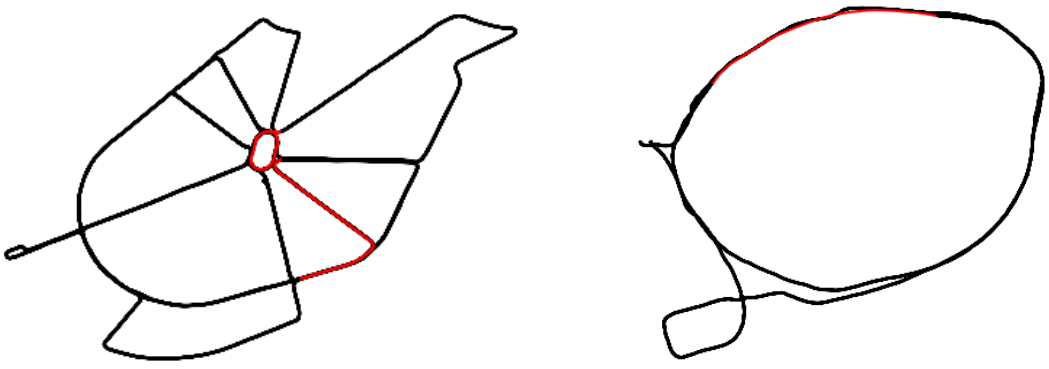

Figure A.1: Example of reference and test sets for VBR [61] *ciampino* and *colosseo*, on the left and on the right respectively. The test samples are shown in red, while black represents the samples used as references.

Scene	Timestamp range for test images
campus_1	1273791462170 – 1324036965410
ciampino_0	3333899690710 – 3361796843540 4320655133829 – 4394998052910
colosseo	2353839583910 – 2442227787890
diag	2824193513930 – 3062070457450
pincio	284343555140 – 707687060340
spagna	213141675270 – 460611290830

Table A.1: Timestamp ranges splitting VBR [61] scenes into reference and test sets. Reference images are all the images taken in timestamps excluded from the reported intervals. Test sets are made of 750 images uniformly sampled from the intervals shown in the table, considering only the left camera.

Scene	Retrieval	$N = 5$		$N = 10$		$N = 20$	
		R@(1°,0.1m)	R@(5°,1m)	R@(1°,0.1m)	R@(5°,1m)	R@(1°,0.1m)	R@(5°,1m)
CAB Hololens	NetVLAD [26]	.378	.567	.446	.629	.481	.667
	AP-GeM [49]	.393	.566	.438	.631	.471	.677
	SALAD [50]	.482	.707	.535	.752	.570	.783
	random	.520	.794	.591	.846	.633	.865
	pose-near	.603	.860	.645	**.889**	.662	**.901**
	pose-coverage	.561	.808	.614	.848	.634	.872
	covisibility	.606	.849	.625	.867	.644	.878
	covisibility-coverage	**.635**	**.868**	**.658**	.885	**.670**	.891
CAB iPad	NetVLAD [26]	.404	.505	.406	.512	.467	.563
	AP-GeM [32]	.396	.492	.416	.515	.434	.555
	SALAD [50]	.429	.551	.446	.551	.475	.601
	random	.454	.621	.515	.654	.575	.689
	pose-near	**.674**	**.856**	**.684**	**.833**	**.654**	**.775**
	pose-coverage	.475	.644	.538	.664	.575	.689
	covisibility	.505	.641	.573	.691	.591	.684
	covisibility-coverage	.523	.656	.591	.699	.591	.717

Table A.2: Localization recalls for LaMAR [4] CAB scene, using the thresholds proposed in the original paper, retrieving a different number N of images.

Scene	Retrieval	$N = 5$		$N = 10$		$N = 20$	
		R@(1°,0.1m)	R@(5°,1m)	R@(1°,0.1m)	R@(5°,1m)	R@(1°,0.1m)	R@(5°,1m)
HGE Hololens	NetVLAD [26]	.401	.571	.443	.569	.472	.622
	AP-GeM [49]	.424	.582	.462	.610	.483	.644
	SALAD [50]	.542	.704	.578	.726	.591	.738
	random	.570	.785	.640	.820	.676	.844
	pose-near	.670	**.875**	.693	**.879**	.709	**.882**
	pose-coverage	.615	.819	.651	.838	.679	.854
	covisibility	.523	.656	.674	.854	.702	.861
	covisibility-coverage	**.685**	.856	**.718**	.869	**.718**	.869
HGE iPad	NetVLAD [26]	.582	.775	.634	.829	.672	.882
	AP-GeM [49]	.512	.747	.588	.817	.659	.859
	SALAD [50]	.655	.913	.712	.941	.728	.943
	random	.613	.880	.697	.933	.729	.947
	pose-near	.725	.937	**.768**	**.962**	**.756**	**.971**
	pose-coverage	.320	.915	.716	.943	.722	.956
	covisibility	.699	.936	.726	.949	.727	.947
	covisibility-coverage	**.731**	**.941**	.747	.952	.752	.960

Table A.3: Localization recalls for LaMAR [4] HGE scene, using the thresholds proposed in the original paper, retrieving a different number N of images.

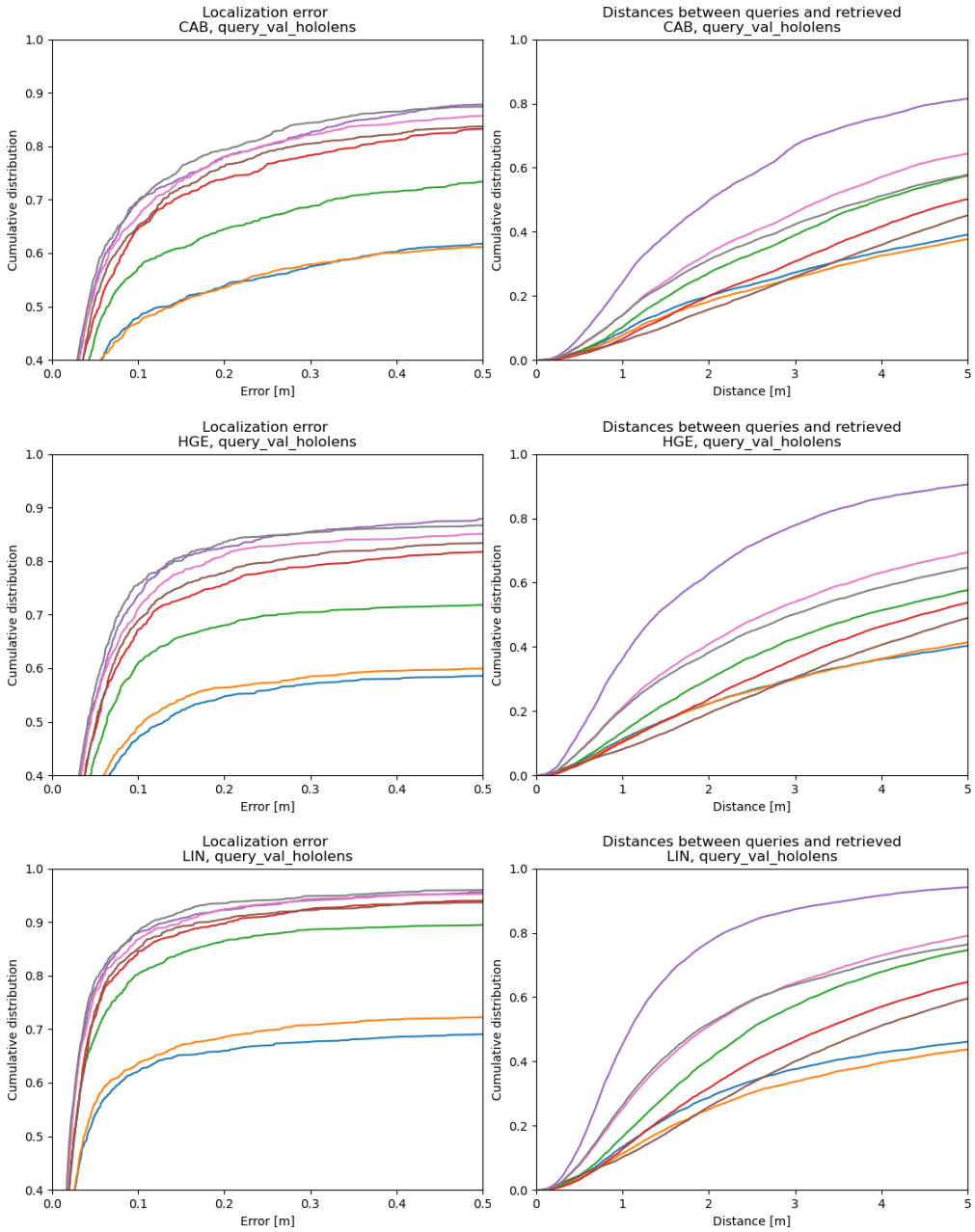

Figure A.2: Localization error and distance between queries and retrievals for Hololens validation images in LaMAR [4]. The comparison is between actual methods, namely NetVLAD [26] (■), AP-GeM [49] (■), SALAD [50] (■), and retrievals exploiting the ground-truths (Section 3.2.1.2): *random* (■), *pose-near* (■), *pose-coverage* (■), *covisibility* (■), *covisibility-coverage* (■). The results are obtained retrieving 10 images.

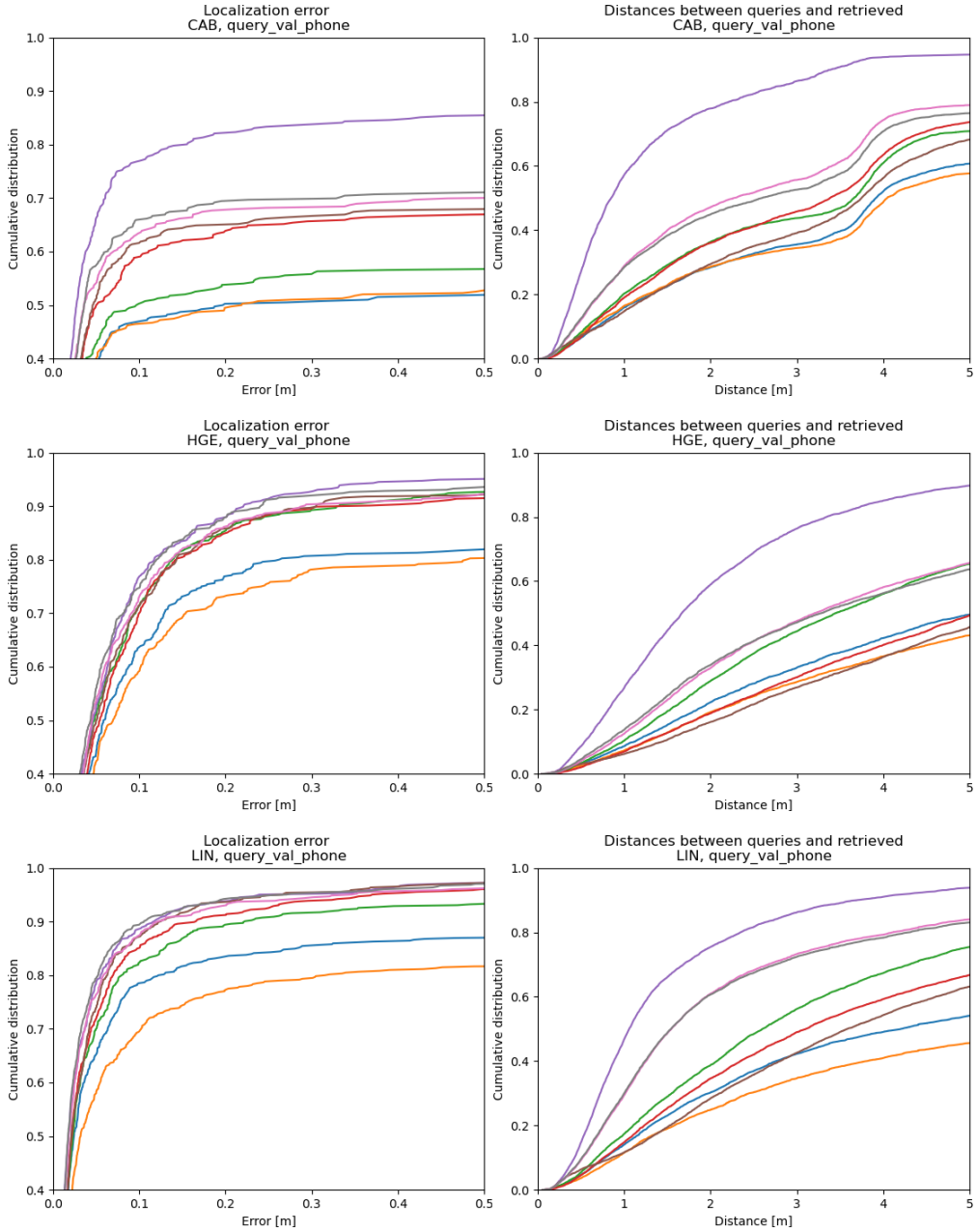

Figure A.3: Localization error and distance between queries and retrievals for iPad validation images in LaMAR [4]. The comparison is between actual methods, namely NetVLAD [26] (■), AP-GeM [49] (■), SALAD [50] (■), and retrievals exploiting the ground-truths (Section 3.2.1.2): *random* (■), *pose-near* (■), *pose-coverage* (■), *covisibility* (■), *covisibility-coverage* (■). The results are obtained retrieving 10 images. The different behavior in CAB is due to the images containing ambiguous and repeated patterns, showing corridors, stairs and classrooms.

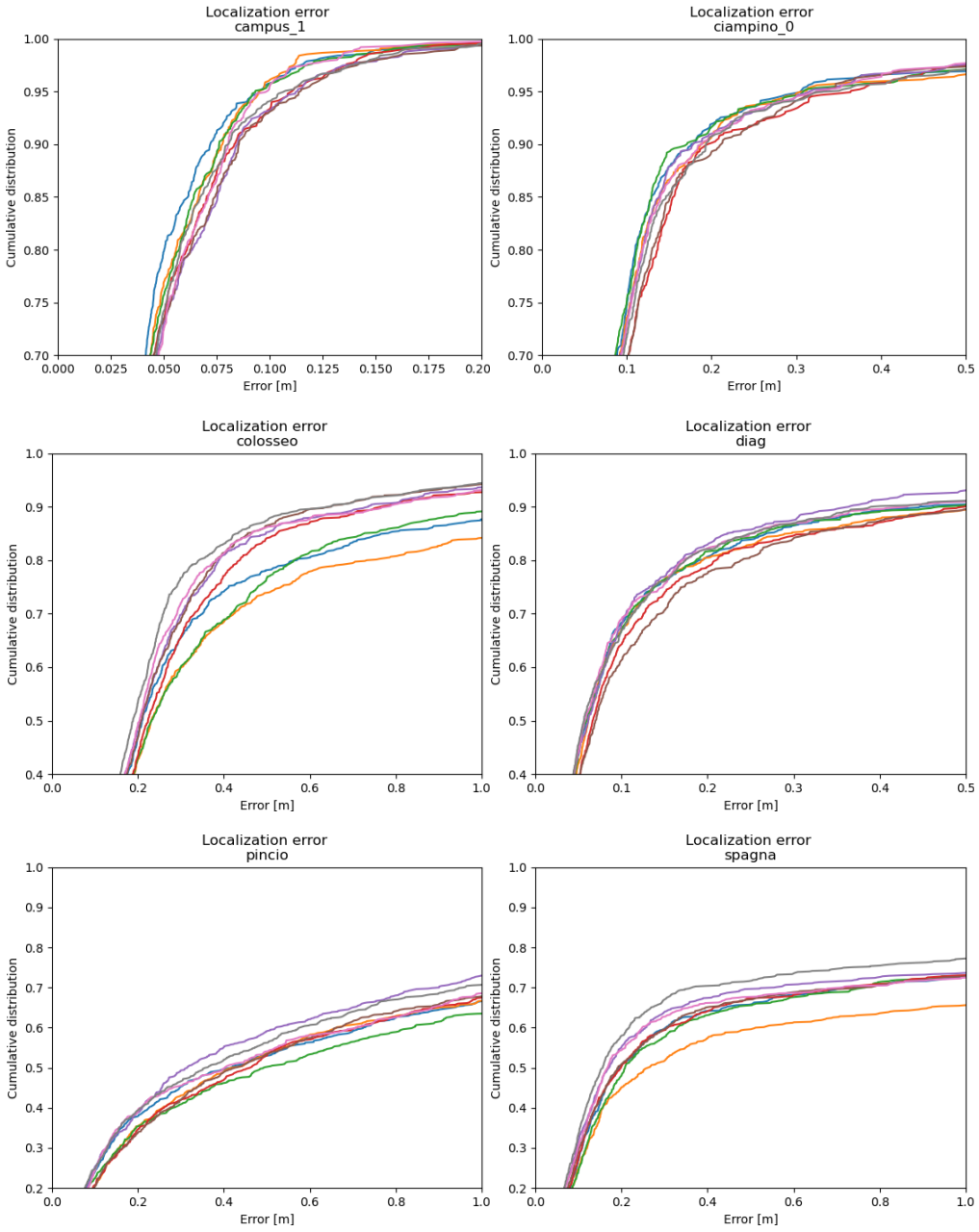

Figure A.4: Localization error for test images in VBR [61]. The comparison is between actual methods, namely NetVLAD [26] (■), AP-GeM [49] (■), SALAD [50] (■), and retrievals exploiting the ground-truths (Section 3.2.1.2): *random* (■), *pose-near* (■), *pose-coverage* (■), *covisibility* (■), *covisibility-coverage* (■). The results are obtained retrieving 10 images.

Scene	Retrieval	$N = 5$		$N = 10$		$N = 20$	
		R@(1°,0.1m)	R@(5°,1m)	R@(1°,0.1m)	R@(5°,1m)	R@(1°,0.1m)	R@(5°,1m)
LIN Hololens	NetVLAD [26]	.566	.662	.603	.699	.629	.720
	AP-GeM [49]	.583	.689	.619	.729	.655	.757
	SALAD [50]	.746	.889	.777	.903	.806	.915
	random	.763	.919	.814	.942	.836	.951
	pose-near	.828	.952	.847	**.963**	.859	.963
	pose-coverage	.779	.921	.821	.938	.836	.952
	covisibility	.821	**.954**	.835	.958	.842	.959
	covisibility-coverage	**.836**	.953	**.854**	**.963**	**.865**	**.964**
LIN iPad	NetVLAD [26]	.727	.843	.774	.874	.785	.898
	AP-GeM [49]	.661	.798	.689	.823	.736	.869
	SALAD [50]	.793	.929	.803	.945	.833	.960
	random	.803	.965	.838	.975	.867	.980
	pose-near	**.873**	**.975**	**.873**	**.977**	**.886**	.980
	pose-coverage	.843	.972	.858	.975	.861	.980
	covisibility	.846	.952	.861	.967	.871	.977
	covisibility-coverage	.854	.955	.883	**.977**	.881	**.982**

Table A.4: Localization recalls for LaMAR [4] LIN scene, using the thresholds proposed in the original paper, retrieving a different number N of images.

corridors, stairs, and classrooms, and thus *pose-near* outperforms everything since it is able to select the correct repeated elements, while methods relying only on covisibility, which is empirically determined with matching points, struggle.

The same conclusion are drawn from Tables A.2, A.3 and A.4, which also compare the effect of the number of retrieved images.

A.3 Qualitative analysis of retrieval

The focus of this section is to qualitatively evaluate the retrieval, to understand if retrieval is intrisincally ill-posed due to the nature of the images itself, or if any improvements seems possible from a human perspective. Some examples of badly localized images are shown in Figure A.5. Two major problems emerge: ambiguity and lack of informativeness. Some images (A.5.**a**) are so uninformative that retrieval is really hard, and even in case of correct retrieval they may be impossible to localize. However, in some cases, there are details that should provide enough distinctiveness, such as the signs in A.5.**b**, the geometric relation between the stairs, the handrail and the colored wall in A.5.**c**, and the picture in the frame in A.5.**d**. Ambiguity may be due to repetitive structures, such as the pattern on the building in A.5.**e**. Again, some details such as the protrusion of the building in A.5.**f** should allow for a proper retrieval and localization.

While the lack of informativeness is hard to face, the needs of retrieval for Visual Localization are different than the typical task of Image Retrieval. In fact, models are trained to be invariant to the viewpoints and to focus on the overall, while, as shown in this qualitative analysis of the negatives, the distinctiveness of details as well as the importance of similar viewpoints are crucial.

Figure A.5: Examples of retrievals for the CAB scene of LaMAR [4] dataset. The retrieval shown is performed with SALAD [50], purposely selecting problematic queries incorrectly located, *i.e,* with more than 40 meters of error. While some queries are uninformative (**a**), some have minor details that should enable correct retrieval and localization (**b**,**c**,**d**). Another problem is ambiguity, such as in case of repetitive patterns (**e**), which is mitigable in case of some geometric peculiarities (**f**).

A.4 Sampling from retrievals

As discussed in Section 3.2.2, it is possible to retrieve more images with standard methods such as SALAD [50], and then selecting a subset of images to perform localization with. In particular, *pose-near* and *covisibility* (Section 3.2.1) are tested on LaMAR [4], as shown in Figure A.6. This approach is able to reduce the gap with respect to the ground-truth upper bounds, and localization using the subset perform on par or better than using the whole starting set. This highlights that the quality of retrieval is more important than the quantity, and that fine tuning existing models or slight alterations of the pipeline are enough to improve results, and major modifications may not be needed to improve Visual Localization.

A.5 Visual Localization with local maps

In some cases, such as large scale dataset, keeping a global map is prohibitive. Thus, local maps are built at query time, using only the image retrieved. The performances of localization using this approach are shown in Figure A.7. Even in this scenario, there is no benefit in retrieving sparse images, as *pose-near* and covisibility approaches have the best results. Furthermore, this approach has at least a 5-10% performance drop with respect to the global maps. Such drop is higher for worst performing methods, while good retrievals are less sensitive to the paradigm change. The local map procedure seem viable only when it is not possible to adopt the standard global approach, possibly adopting robust retrievals. This approach may benefit by retrieving more images, building better local reconstructions, at the cost of exponentially increase runtime.

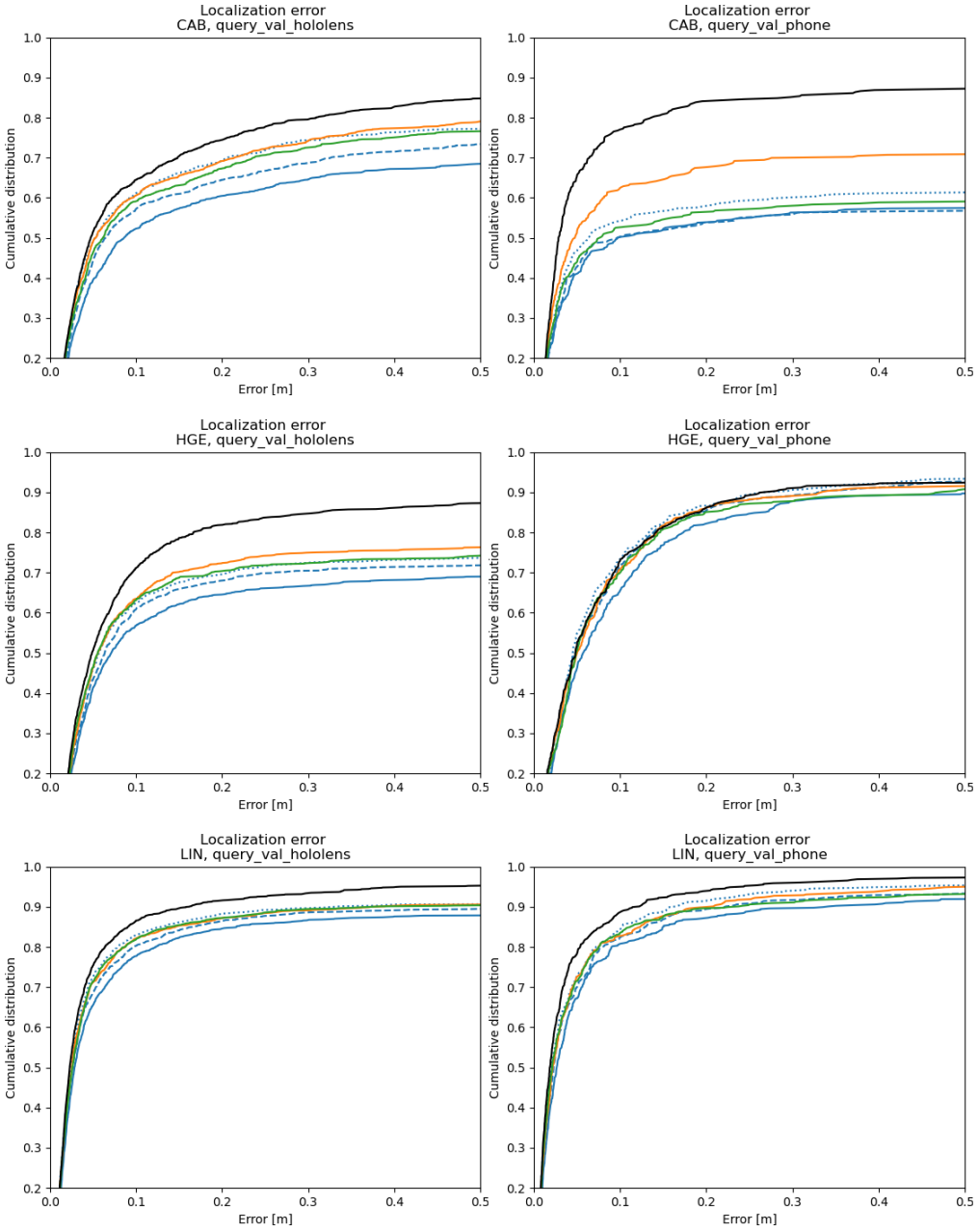

Figure A.6: Localization error in LaMAR [4]. The comparison is between plain SALAD [50] (■) with 5, 10, and 20 retrieved images (respectively, solid, dashed and dotted lines), the ground-truth baseline obtained with *pose-near* (Section 3.2.1) (■), and sampling 5 images among the 20 retrieved with SALAD using *pose-near* (■) and *covisibility* (■).

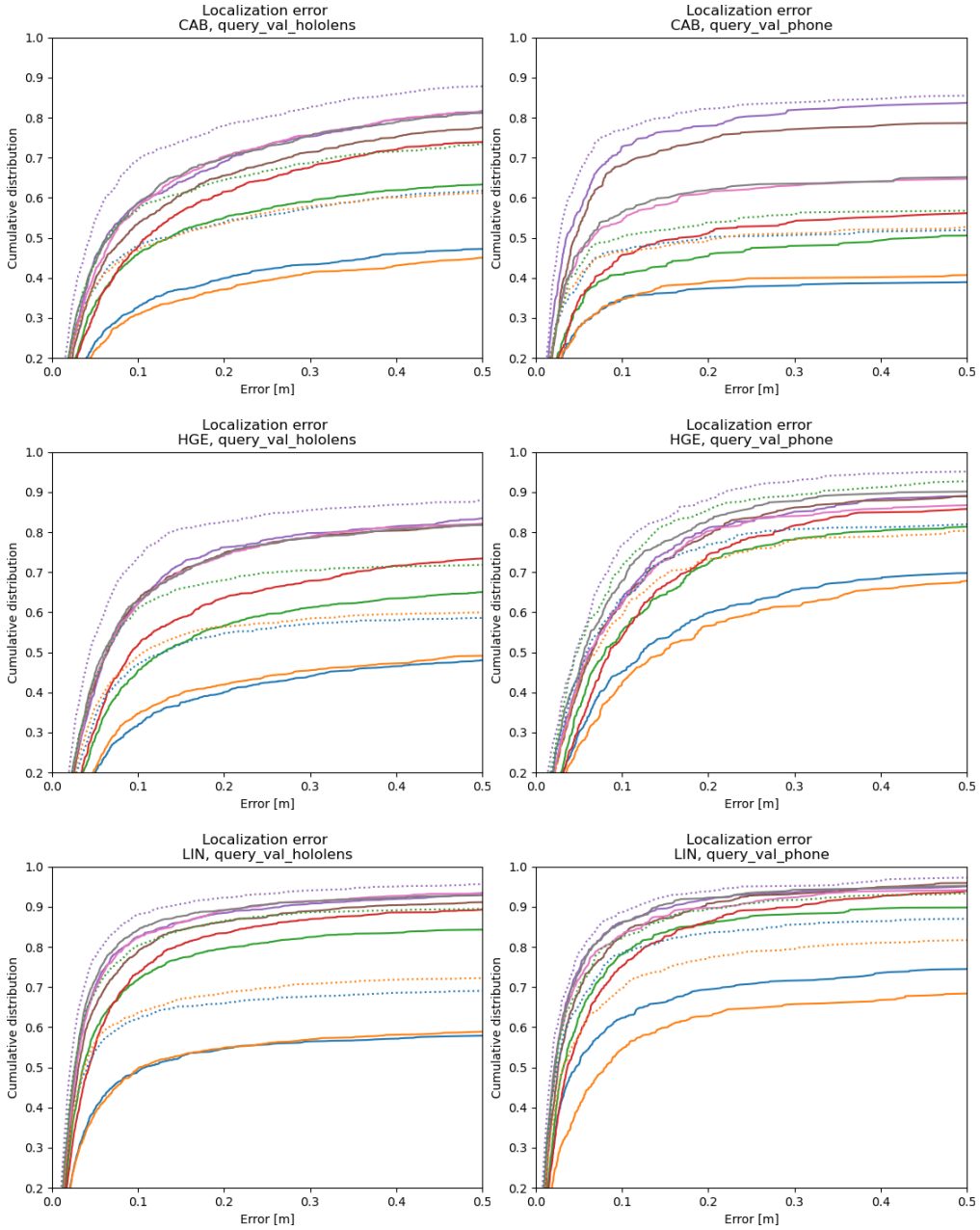

Figure A.7: Localization error for validation images in LaMAR [4] using local maps. The comparison is between actual methods, namely NetVLAD [26] (■), AP-GeM [49] (■), SALAD [50] (■), and retrievals exploiting the ground-truths (Section 3.2.1.2): *random* (■), *pose-near* (■), *pose-coverage* (■), *covisibility* (■), *covisibility-coverage* (■). For *pose-coverage* the images are filtered to be within an adaptive radius from the query, starting from 3m and increased by 50% iteratively if not enough images are kept. Dotted lines are obtained using global maps, to ease comparison between the approaches. The results are obtained retrieving 10 images.

Bibliography

[1] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. "Brief: Binary robust independent elementary features". In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer. 2010, pp. 778–792 (cit. on p. 2).

[2] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. "Lost in quantization: Improving particular object retrieval in large scale image databases". In: *2008 IEEE conference on computer vision and pattern recognition*. IEEE. 2008, pp. 1–8 (cit. on p. 2).

[3] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. "From coarse to fine: Robust hierarchical localization at large scale". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 12716–12725 (cit. on p. 2).

[4] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. "Lamar: Benchmarking localization and mapping for augmented reality". In: *European Conference on Computer Vision*. Springer. 2022, pp. 686–704 (cit. on pp. 3, 19, 23, 25, 26, 28, 30, 31, 34, 38–40, 42–46).

[5] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. "Deep visual geo-localization benchmark". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5396–5407 (cit. on pp. 3, 23).

[6] Martin Humenberger, Yohann Cabon, Noé Pion, Philippe Weinzaepfel, Donghwan Lee, Nicolas Guérin, Torsten Sattler, and Gabriela Csurka. "Investigating the role of image retrieval for visual localization: An exhaustive benchmark". In: *International Journal of Computer Vision* 130.7 (2022), pp. 1811–1836 (cit. on pp. 3, 23).

[7] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. "Are large-scale 3d models really necessary for accurate visual localization?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1637–1646 (cit. on pp. 3, 23, 29).

[8] Chris Harris, Mike Stephens, et al. “A combined corner and edge detector”. In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244 (cit. on p. 5).

[9] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60 (2004), pp. 91–110 (cit. on pp. 5, 6).

[10] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. “Speeded-up robust features (SURF)”. In: *Computer vision and image understanding* 110.3 (2008), pp. 346–359 (cit. on p. 6).

[11] Relja Arandjelović and Andrew Zisserman. “Three things everyone should know to improve object retrieval”. In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 2911–2918 (cit. on pp. 6, 19).

[12] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. “Lift: Learned invariant feature transform”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer. 2016, pp. 467–483 (cit. on p. 6).

[13] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. “Large-scale image retrieval with attentive deep local features”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 3456–3465 (cit. on p. 7).

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012) (cit. on pp. 7, 16).

[15] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014) (cit. on pp. 7, 13, 16).

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 7, 17).

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. “Superpoint: Self-supervised interest point detection and description”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 224–236 (cit. on pp. 7, 13, 14, 25).

[18] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. “D2-net: A trainable cnn for joint description and detection of local features”. In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2019, pp. 8092–8101 (cit. on p. 7).

[19] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. “R2d2: Reliable and repeatable detector and descriptor”. In: *Advances in neural information processing systems* 32 (2019) (cit. on p. 7).

[20] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "Superglue: Learning feature matching with graph neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4938–4947 (cit. on p. 7).

[21] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. "Lightglue: Local feature matching at light speed". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 17627–17638 (cit. on pp. 7, 13–15, 22, 25).

[22] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. "Aggregating local image descriptors into compact codes". In: *IEEE transactions on pattern analysis and machine intelligence* 34.9 (2011), pp. 1704–1716 (cit. on p. 8).

[23] Tommi Jaakkola and David Haussler. "Exploiting generative models in discriminative classifiers". In: *Advances in neural information processing systems* 11 (1998) (cit. on p. 8).

[24] Florent Perronnin and Christopher Dance. "Fisher kernels on visual vocabularies for image categorization". In: *2007 IEEE conference on computer vision and pattern recognition*. IEEE. 2007, pp. 1–8 (cit. on p. 8).

[25] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. "Aggregating local descriptors into a compact image representation". In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 3304–3311 (cit. on pp. 8, 15).

[26] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. "NetVLAD: CNN architecture for weakly supervised place recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5297–5307 (cit. on pp. 8, 15, 17, 23, 25, 26, 30–32, 34, 37–42, 46).

[27] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. "From generic to specific deep representations for visual recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 36–45 (cit. on pp. 8, 16).

[28] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. "Particular object retrieval with integral max-pooling of CNN activations". In: *arXiv preprint arXiv:1511.05879* (2015) (cit. on pp. 8, 16).

[29] Artem Babenko and Victor Lempitsky. "Aggregating deep convolutional features for image retrieval". In: *arXiv preprint arXiv:1510.07493* (2015) (cit. on pp. 8, 16).

[30] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. "Deep image retrieval: Learning global representations for image search". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer. 2016, pp. 241–257 (cit. on pp. 9, 17).

[31] Filip Radenović, Giorgos Tolias, and Ondřej Chum. "CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 3–20 (cit. on p. 9).

[32] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 539–546 (cit. on pp. 9, 38).

[33] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. "Multi-similarity loss with general pair weighting for deep metric learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5022–5030 (cit. on pp. 9, 18).

[34] Kun He, Yan Lu, and Stan Sclaroff. "Local descriptors optimized for average precision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 596–605 (cit. on pp. 9, 17).

[35] Amit Singhal et al. "Modern information retrieval: A brief overview". In: *IEEE Data Eng. Bull.* 24.4 (2001), pp. 35–43 (cit. on p. 9).

[36] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003 (cit. on p. 10).

[37] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395 (cit. on pp. 10, 19, 23, 25, 29, 30).

[38] Christopher Sweeney, Tobias Hollerer, and Matthew Turk. "Theia: A fast and scalable structure-from-motion library". In: *Proceedings of the 23rd ACM international conference on Multimedia*. 2015, pp. 693–696 (cit. on p. 10).

[39] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. "Openmvg: Open multiple view geometry". In: *Reproducible Research in Pattern Recognition: First International Workshop, RRPR 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers 1*. Springer. 2017, pp. 60–74 (cit. on p. 10).

[40] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. "Global structure-from-motion revisited". In: *European Conference on Computer Vision*. Springer. 2024, pp. 58–77 (cit. on pp. 10, 18, 34).

[41] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. "Bundle adjustment—a modern synthesis". In: *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Springer. 2000, pp. 298–372 (cit. on pp. 10, 11).

[42] Noah Snavely, Steven M Seitz, and Richard Szeliski. "Photo tourism: exploring photo collections in 3D". In: *ACM siggraph 2006 papers*. 2006, pp. 835–846 (cit. on p. 10).

[43] Changchang Wu. "Towards linear-time incremental structure from motion". In: *2013 International Conference on 3D Vision-3DV 2013*. IEEE. 2013, pp. 127–134 (cit. on p. 10).

[44] Johannes L Schonberger and Jan-Michael Frahm. "Structure-from-motion revisited". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4104–4113 (cit. on pp. 10, 18, 23, 25, 30, 33, 34).

[45] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1874–1883 (cit. on pp. 13, 14).

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755 (cit. on p. 13).

[47] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. "Roformer: Enhanced transformer with rotary position embedding". In: *Neurocomputing* 568 (2024), p. 127063 (cit. on p. 15).

[48] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017) (cit. on p. 15).

[49] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. "Learning with average precision: Training image retrieval with a listwise loss". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5107–5116 (cit. on pp. 15, 16, 23, 25, 26, 30–32, 37–42, 46).

[50] Sergio Izquierdo and Javier Civera. "Optimal transport aggregation for visual place recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 17658–17668 (cit. on pp. 15, 17, 22, 23, 25, 26, 28–32, 37–46).

[51] Filip Radenović, Giorgos Tolias, and Ondřej Chum. "Fine-tuning CNN image retrieval with no human annotation". In: *IEEE transactions on pattern analysis and machine intelligence* 41.7 (2018), pp. 1655–1668 (cit. on pp. 15, 16).

[52] Relja Arandjelovic and Andrew Zisserman. "All about VLAD". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2013, pp. 1578–1585 (cit. on p. 16).

[53] Google LLC. *Google Street View Time Machine*. 2007. URL: `https://www.google.com/streetview/` (visited on 02/05/2025) (cit. on pp. 16, 18).

[54] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge J Belongie. "Integral channel features." In: *Bmvc*. Vol. 2. 3. London, UK. 2009, p. 5 (cit. on p. 16).

[55] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115 (2015), pp. 211–252 (cit. on p. 17).

[56] Maxime Oquab et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023) (cit. on p. 17).

[57] Marco Cuturi. "Sinkhorn distances: Lightspeed computation of optimal transport". In: *Advances in neural information processing systems* 26 (2013) (cit. on p. 18).

[58] Richard Sinkhorn and Paul Knopp. "Concerning nonnegative matrices and doubly stochastic matrices". In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348 (cit. on p. 18).

[59] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguere. "Gsv-cities: Toward appropriate supervised visual place recognition". In: *Neurocomputing* 513 (2022), pp. 194–203 (cit. on p. 18).

[60] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. *Ceres Solver*. Version 2.2. Oct. 2023. URL: `https://github.com/ceres-solver/ceres-solver` (cit. on p. 19).

[61] Leonardo Brizi, Emanuele Giacomini, Luca Di Giammarino, Simone Ferrari, Omar Salem, Lorenzo De Rebotti, and Giorgio Grisetti. "VBR: A Vision Benchmark in Rome". In: *arXiv preprint arXiv:2404.11322* (2024) (cit. on pp. 19, 20, 23, 27, 37, 38, 41).

[62] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *kdd*. Vol. 96. 34. 1996, pp. 226–231 (cit. on p. 21).