POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

Understanding the Role of Visual Place Recognition for 3D Reconstruction

Relatori
Prof. Carlo Masone
Gabriele Berton
Gabriele Trivigno

Candidato
Valerio Gallo

Anno Accademico 2023-2024

Abstract

Structure from Motion (SfM) is the task of building a 3D reconstruction of the surrounding space, extracting information from 2D images. Visual Place Recognition (VPR), on the other hand, is the task of recognising and matching images depicting the same place, possibly from different points of view. A crucial step in SfM pipelines, allowing both to increase efficiency and boost performance, is matching similar images with VPR algorithms before the reconstruction is initialised. This thesis aims at investigating the role of VPR in the process of building a 3D reconstruction, pointing out the features that a place recognition method should have to work fine in this task. This is achieved carrying out numerical experiments over images, sourced from the Internet, of three large scale scenes. An original evaluation strategy, based on point reprojection on a set of test images, is proposed and used to compare performances of various VPR methods and strategies in a Structure from Motion pipeline.

Contents

Chapter 1

Introduction

Recent years have seen an exponential growth of the trend of Artificial Intelligence. To be clear, this area has been under development for several years now, but recently it has progressively become part of our everyday life. In general the field of Artificial Intelligence aims at building an intelligent system capable of receiving information from the world, processing it and utilizing it to make decisions and interacting again with the world. In other words its goal is to replicate and enhance human-like intelligence in machines, enabling them to perform tasks that were previously thought to require human cognitive abilities. Clearly such a general definition can be applied to several contexts, such as healthcare, finance, robotics, natural language processing and many others.

Such a system has to overcome many challenges. For example the knowledge it acquires has to be stored in an efficient way, to allow fast consultation of vast databases. Then the way the system interacts with the world can be extremely varied, some models are designed for a specific task, while others are more general purpose ones, and tend to work at an higher level. But one of the more complex, and in my opinion interesting, steps of the pipeline is when the system has to attain information from the environment. Focusing on this point leads us to the non trivial question of how the machine perceives the world. This is typically done through the usage of sensors that try to emulate in some way the human senses. But it is important to consider that the information to acquire can be very heterogeneous, as well as noisy and chaotic.

One of the most important, but maybe also of the most complex, senses that Artificial Intelligence has to try to replicate is the human vision. Fortunately we live in an era where there is an abundance of images and videos available, and it is easy and affordable to acquire new ones. However, as I said, this large amount of data is often chaotic and it becomes crucial to understand the better way to "feed" the system with visual data. The branch of Artificial Intelligence that deals with this is called Computer Vision. In this field two primary technologies are used: a type of machine learning called deep learning and convolutional neural networks (CNN). Machine learning employs algorithmic models that enable a computer to autonomously learn the context of visual data. When sufficient information is input into these models, the computer “looks at” the data and learns to distinguish one image from another on its own. These algorithms allow the machine to learn autonomously, without being explicitly programmed to recognize images. A CNN in this context is crucial to assist a machine learning or deep learning model in breaking down images into pixels, which are then assigned tags or labels. These labels are used to perform convolutions and make predictions based on what is seen. The neural network performs convolutions and checks the accuracy of its predictions over a series of iterations until the predictions begin to align. As a result, it can recognize or perceive images similarly to humans. Similar to how a person distinguishes an image from a distance, a CNN initially discerns sharp outlines and simple shapes, gradually adding details as it iteratively refines its predictions. A CNN is utilized for analyzing individual images. Conversely, a recurrent neural network (RNN) is often employed similarly in video applications to help computers understand how images across frames in a sequence are interconnected. Obviously such an approach requires large amount of data. As it is easy to imagine there are countless computer vision tasks being studied.

Many applications for example require a three-dimensional model of the surrounding space. It is sufficient to think about robotics or autonomous driving vehicles. This can be achieved through sensors capable of perceiving the 3D space, but unfortunately these are often expensive and not so easy to find. Humans perceive a great deal of information about the three-dimensional structure of the environment by moving around it. When the observer moves, objects around them move in a different way depending on their distance from the observer. This principle is known as motion parallax, and from this, depth information can be earned and used to generate an accurate 3D mental representation of the surrounding world. In addition, even if we are not moving, our brain receives images from two different points of view, since we have two eyes, and this facilitates the creation of a 3D model of the space. A class of Artificial Intelligence technologies that try to replicate this human capability goes under the name of Structure from Motion (SfM). As we will further explore in the following chapters, SfM algorithms aim at building a 3D model from easily accessible 2D images.

To achieve a precise and efficient reconstruction it is often necessary to group the many various images into similarity clusters. This task is entrusted to Visual Place Recognition algorithms. Their goal is to estimate the geographical position where a given picture was taken by comparing it to a large database of images with known locations. In the SfM context anyway they are used to look for images that are similar in some sense. Indeed the concept of “similarity” leaves room for different interpretations, and distinct algorithms can give different results. It may even be that models that work better for Visual Geo-Localization, perform less effectively in a SfM pipeline. The aim of this thesis is to focus on how this algorithms work and to understand, by means of numerical experiments, which methods and configurations perform better than others in this context.

The remaining chapters will be structured as follows. Chapter 2 will delve into Visual Place Recognition in details, with particular attention to the approaches used in the experiments. Chapter 3 will shift its focus to Structure from Motion methods, discussing both underlying theoretical principles and a state-of-the-art algorithm called COLMAP. Chapter 4 will present the problem setting and the numerical outcomes of the experiments. The results of extensive testing will be shown and discussed, with the aim of understanding what are the best performing configurations of the Visual Place Recognition step in a SfM pipeline.

Chapter 2

Visual Place Recognition

In this chapter the tasks of Visual Place Recognition (VPR) will be presented, following this general structure. First of all, there will be a broad discussion on the most relevant approaches that, over the years, have proven to be more successful. Then a more detailed analysis will explore some of the current state-of-the-art models, and in particular the ones used in the numerical experiments: NetVLAD and Cosplace. Many general concepts of deep learning will be assumed as known and not delved into details, since a full coverage of this topics would be unfeasible, other than pointless. However, a reference to useful literature material will be given when they are first mentioned.

2.1 Defining the problem

What is Visual Place Recognition? Bellavia et al. [2024] state that VPR addresses the question of "given an image of a place, can a human, animal, or robot decide whether or not this image is of a place it has already seen?". It is easy to infer that such a capability is of crucial importance in tasks like localization and navigation, which recently became ever more important with the advent of Artificial Intelligence in autonomous cars, mobile robots, and reality augmentation platforms. Such a variety of uses and application domains translates to a rich research panorama where VPR is studied by different communities (computer vision, robotics, machine learning) and with different problem settings. For instance, in computer vision VPR is often studied as the task of recognizing the location of a single image. In robotics, VPR algorithms can typically leverage streams of heterogeneous data (e.g., videos, pointclouds, odometry, etc.) as well as some knowledge of the motion of the robot. Moreover, in robotics there is a stronger emphasis on computational efficiency and real-time execution. Focusing on a Computer Vision setting, Visual Place Recognition (VPR), also known as Visual Geo-Localization (VG) or image localization, can be defined as the task of recognizing the "place" depicted in a given image by comparing it to a large dataset of images, whose locations are known a priori. Even the apparently trivial definition of place may change depending on the task: a place could be denoted by the name of a landmark, a GPS coordinate or even a 6 DoF pose with respect to a frame of reference. However, as mentioned before, in this work I will focus on this task when it is a step of a Structure from Motion pipeline. In this scenario it would be more appropriate to speak of *Image Retrieval*. This refers to the general problem of retrieving relevant images from a large database. VPR is commonly cast as an image retrieval problem that involves a nearest neighbor search of compact global descriptors or cross-matching of local descriptors. Figure 2.1, taken from Masone and Caputo [2021], illustrates how image retrieval works in general. With regards to solving the nearest neighbor search problem, VPR and image-retrieval systems face similar challenges. However, the underlying goals differ between the two areas. For image retrieval, similarity criteria could be based on semantic categories such as a product category or an environmental condition category. However, with the additional context of being a “place”, VPR deviates from the process of merely retrieving a similar image. The notion of similarity in VPR is constrained to matching spatial information, where images captured from the same place would be considered a true match even if environmental conditions are dissimilar (e.g. day and night images).

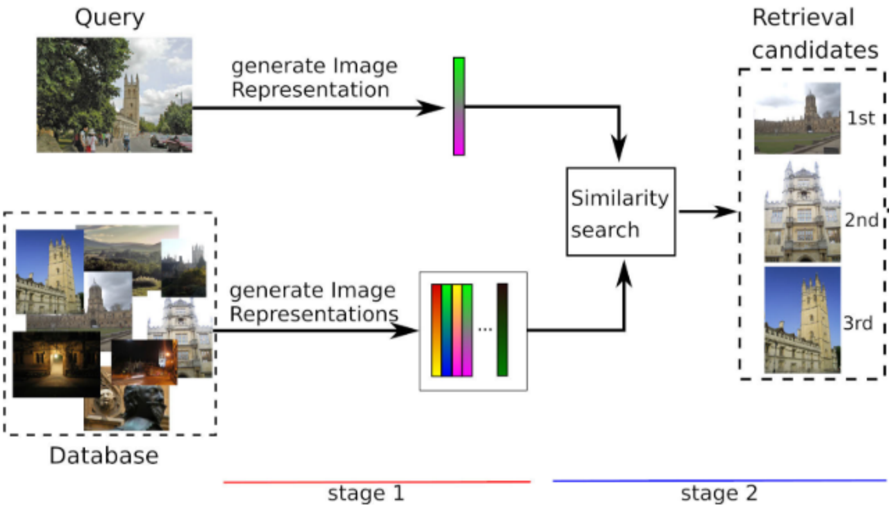


Figure 2.1: Visual place recognition is commonly formulated as an image retrieval problem. The known places are collected in a database and a new image to be localized is called query. The place retrieval is performed in two logical stages. 1) In the first stage, vector representations are generated for the query and the database images. 2) The representation of the query is compared to those of the database images, to find the most similar ones (here only the top 3 are shown).

In both formulation, the prior knowledge of the places of interest for the task is represented as a collection of images (database). Each image in the database is tagged with an identifier of its location, that could be the name of a landmark or a GPS coordinate, depending on the final purpose. A deep model (usually a convolutional neural network) is then used to extract a vector of descriptors from each database image. These contain all the information that the network considers relevant in the picture. When a new picture needs to be localized (query), the model extracts descriptors of the query in the same way. The place recognition system later searches through the database for images that are similar to it. At this point the concept of similar images comes down to an high value of some similarity score computed over the numerical descriptors, that can be as simple as the *euclidean distance* or a more complex *cosine similarity.* If similar pictures are found, their tagged locations are used to infer the location of the query. In a geo-localization setting, for each query, the k best scoring images are considered and their coordinates are used as estimates for the query position. If they are within a fixed error threshold, then the prediction is considered correct. The most common metric is *recall@k*, which represents the percentage of queries for which at least one of the k best estimates is correct. This metric can be used not only for the evaluation, but also in previous stages for the training of the network.

The retrieval paradigm is applied in many other fields of deep learning, but in VPR it faces some unique challenges strictly related to the nature of the task:

1. *Complexity of the image*: the scenes in which a place appears are complex and there is not a single, identifying object that can be exploited. In most cases the information is scattered across some of the many elements of the image, that could be static or even dynamic. Some of it may even be hidden behind occlusions and not available to the model.

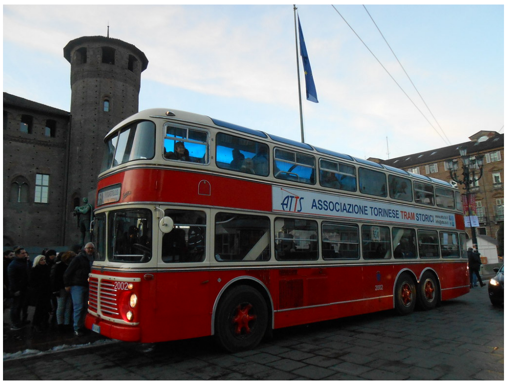


Figure 2.2: This picture is an example of a dynamic obstruction. The bus obscures most of the scene. A retrieval method would probably match this image with other bus pictures, regardless of the place they have been taken.

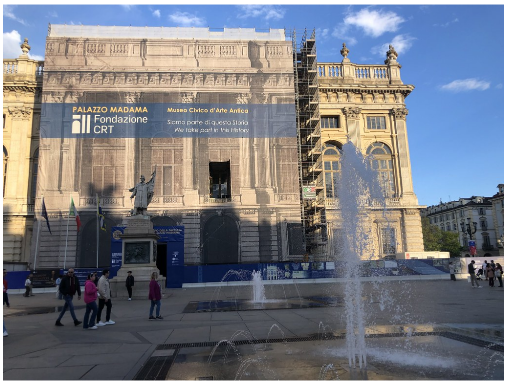

PALAZZO MADAMA
Fondazione CRT
Museo Civico d'Arte Antica
Siamo parte di questa Storia
We take part in this History

Figure 2.3: Here an example of a construction site covering most of the facade of the building. Note also the fountain stream in the foreground, which could trick VPR algorithms.

2. *Point of view*: the same scene can be depicted from many different points of view and the resulting images have different appearances. There is no guarantee that the

database contains a positive match with the same viewpoint of the query.

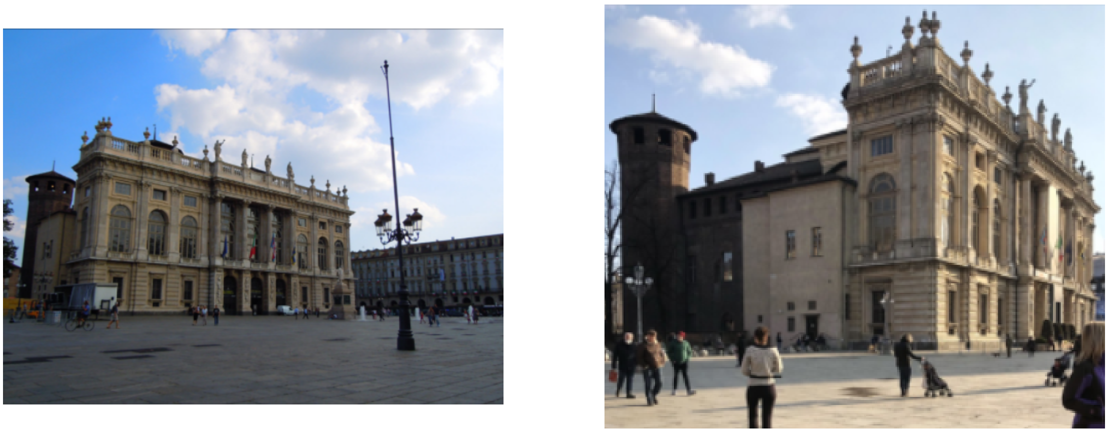


Figure 2.4: These two images depict the same building from different points of view. The change of perspective adds new elements, such as elements of the second facade, and changes the geometry of the building.

3. *Variable conditions*: in real world scenarios the scene conditions are extremely dynamic and can change rapidly. This happens not only because of natural illumination variations, such as the day/night cycle, seasons and weather, but also because of the presence of physical objects, such as temporary construction sites, vehicles and pedestrians.

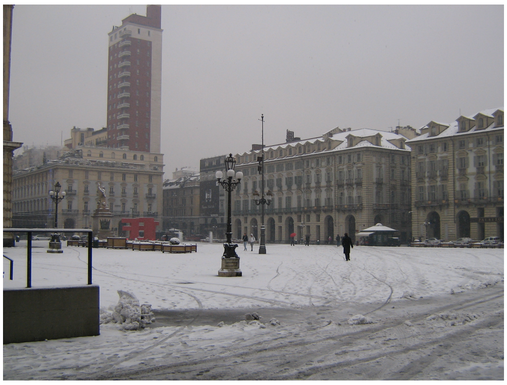


Figure 2.5: This picture is an example of conditions variation due to metereological factors. The snow not only changes colors, but can also hide some distinctive elements and change the shape of the floor.

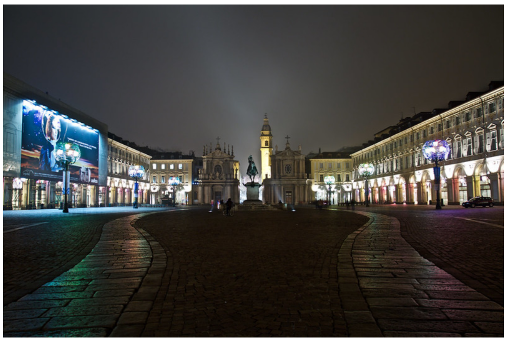


Figure 2.6: Here an example of illumination change. In night pictures some elements are difficult to distinguish, and others are highlighted by artificial illumination. In this case, in addition, the illumination is part of a temporary installation.

Figures from 2.2 to 2.6 are taken from Flickr, and in particular from the Turin scene (see section 4.2).

2.2 History of Visual Place Recognition

Early studies in Visual Place Recognition (VPR) can be found in the field of visual SLAM (Simultaneous Localization And Mapping), mainly correlated to robotics (Durrant-Whyte and Bailey [2006]). In the robotics literature, VPR has historically been called *loop closure detection*. It gained more prominence in the field as the earlier metric SLAM methods based on global and local bundle adjustment techniques, which could only handle limited-size environments, were complemented by topological SLAM techniques based on bag-of-words approaches (Galvez-Lòpez and Tardos [2012]), such as FAB-MAP (Cummins and Newman [2008]). In addition to its relevance within SLAM pipelines, VPR also remains a crucial component of localization-only pipelines where a prior map is available. Early VPR research primarily focused on place recognition under constant or slightly varying environmental conditions. Addressing appearance changes due to more severe condition changes, such as day-night cycles or seasonal shifts, emerged in the late 2000s. These methods usually relied on local feature matching (Valgren and Lilienthal [2010]). A local feature descriptor analyzes only a patch of the image, highlighting patterns that differ from its neighborhood. These patches can be densely sampled, but in visual place recognition they are generally originated from a sparse detector that identifies points of interest (keypoints), such as SIFT (Lowe [1999]) and SURF (Bay et al. [2008]). Two images can thus be compared by analyzing pairwise correspondences among their respective descriptors, however this approach is not effective and hardly scalable to a database-wide search. The possible solution is that for searching the database the images should be compared by analyzing the statistics of their descriptors, rather than matching them on an individual basis. This idea was pioneered by Sivic and Zisserman [2003] who adopted the Bag-of-Words (BoW) approach for image retrieval. In this method the descriptors are quantized in clusters, based on a codebook of visual words, and the image representation is then obtained as the histogram of the assignment of all image descriptors to visual words, weighted according to the “term frequency – inverse document frequency” (tf-idf), to give importance to features appearing many times in the specific image, but not so often over the database. Following these footsteps, other representations based on the quantization of local descriptors have been proposed. Jegou and Zisserman [2014] observe that methods that create a single vector representation from local feature descriptors can be splitted in an embedding step, that individually maps each vector to a higher-dimensionality space, and an aggregation step, that generates a single representation from the mapped vectors. The reason behind the embedding step is to improve the distinctiveness of the individual features and suppress false positives. Notable examples of such methods are Fisher Vectors (Perronnin et al. [2010]) and VLAD (Jegou et al. [2010], Arandjelovic and Zisserman [2013]). While the aggregation of local feature descriptors allows to obtain a single vector representation of an image, this can also be done directly using global feature descriptors, i.e., descriptors that encode holistic properties of the scene. Since they process the image as a whole, global descriptors do not require a detection phase, thus being less expensive to compute. Examples of global descriptors are HOG (Dalal and Triggs [2005]) and Gist (Oliva and Torralba [2006]). Compared to the representation from local descriptors, global descriptors are less robust to viewpoint changes and occlusions. In 2014, the use of deep learning, and in particular of CNNs, for VPR (Chen et al. [2014]) emerged as a way to handle challenging data and has since proven effective in changing environments. Moreover, it has been shown that CNNs can learn generic features that are, to some extent, transferable to other visual tasks. These findings have also inspired the application of deep learned representations to image retrieval, where they have surpassed the performance achieved with previous shallow methods. From the early studies, which demonstrated that the vector of activations of a fully connected (FC) layer of a classification network could be effectively used for retrieval, it became soon clear that the information extracted by a fully connected layer of a CNN is akin to a global descriptor: it is not robust to the presence of distractors or occlusions and lacks invariance to translation and scale. In addition, fully connected representations are limited by the fixed input size and by requiring large numbers of parameters. This limitations of FC representations inspired researchers to investigate the generation of image representations directly from the output of convolutional layers. More specifically, the approach of Babenko et al. [2014] was to take the $H \times W \times C$ tensor produced by a convolutional layer of the network, where H is the height of the tensor, W is its width and C is the number of channels, and flatten it as a vector. This vector is then normalized and used as image representation. Despite the interesting use of the convolutional feature maps, the results achieved with this simple method are not far off from those obtained with fully connected representations. Intuitively, simply flattening the feature maps of a convolutional layer does not take full advantage of the spatial information contained therein. This consideration has guided the development of the current state-of-the-art representations for place retrieval. Rather than collapsing the $H \times W \times C$ features extracted from a convolutional layer to a vector, they can be considered as a $H \times W$ grid of C-dimensional feature descriptors, each one having a limited receptive field over the image. Namely, the output of a convolutional layer can be assimilated to a set of densely extracted local descriptors. These dense descriptors can then be aggregated in a single vector representation and then compared using a similarity function. Several studies demonstrated the applicability of classic encodings to these dense convolutional descriptors, e.g., VLAD (Paulin et al. [2015]) and BoW (Mohedano et al. [2016]). Moving further, researchers have proposed aggregation modules that can be plugged on top of a CNN and allow end-to-end learning. In Arandjelovic et al. [2018] it is introduced NetVLAD, a layer that implements the VLAD embedding and aggregation with differentiable operations, that will be deeper discussed in 2.4. Other researchers have shown that convolutional features from mid or late layers, unlike shallow non-learned features, can be successfully aggregated and compared without embedding. Babenko and Lempitsky [2015] show that for shallow hand-crafted features like SIFT, the embedding step is fundamental to improve their discriminativity. However, they argue that raw convolutional features have a higher discriminative capability and therefore they can be pooled together with simpler schemes, thus providing not only a leaner pipeline and, in many cases, more compact representations, but also improving performance. Namely, an image representation can be generated by summarizing the statistics of the convolutional features through a pooling approach. Some examples of simple pooling strategies are *max-pooling* and *sum-pooling*. The authors of Babenko and Lempitsky [2015] observe that max-pooling is more invariant to scale changes, whereas sum-pooling is less sensitive to distractors in the feature maps. The current state-of-the-art pooling methods are R-MAC (Radenović et al. [2019]) and GeM (Schubert et al. [2021]). In the former a max pooling is applied to a number of patches randomly sampled from the feature maps generated by the convolutional network. The results can be seen as regional descriptors, and are finally summed and L^2-normalized, in order to keep their dimensionality low. The latter, instead, implements the generalized mean operator, which is differentiable, hence the mean parameter can be learned during training. Lately, in addition to images and image descriptors, VPR research has also explored the use of additional information, such as sequences, intra-set similarities, weak GPS signals, or odometry, to improve performance (Schubert et al. [2021]), but I will not focus on these methods in this thesis.

2.3 A generic pipeline

This section outlines a generic pipeline for Visual Place Recognition, using a modern deep learning approach. Let it be clear that the following steps do not refer to a specific method, but represent a general backbone that can be extended or modified in each specific pipeline. This section is intended to give an idea of how these algorithms work and to set a proper notation.

1. **Inputs:** Two sets of images serve as the input in a VPR pipeline: the database set DB and the query set Q. The DB set represents a map of known places through images with known location. The query set Q, on the other hand, is the "live view", often recorded by a different platform than DB or after DB. Both sets will have a geographical overlap and share some or all seen places. The pipeline produces matching decisions, meaning that for each query image $I_j \in Q$, one or more database images $I_i \in DB$ can be associated. There are different VPR problem categories: using just a query set Q (single-session VPR) or using both the DB and Q sets (multi-session VPR). Also, the image sets can either be specified before processing (batch VPR) or grow during an online run (online VPR).

2. **Image-wise descriptor computation:** Image descriptors are abstractions of images that extract information from raw pixels in order to be more robust against changes in appearance and viewpoint. As already stated, there exist two primary types of image descriptors:

   - *Global descriptors* represent an image $I_i \in DB, Q$ with a single vector $d_i \in \mathbb{R}^d$. This allows for efficient pairwise descriptor comparisons with low runtimes. Note that when exhaustive k-nearest neighbor search (kNN) is used to obtain the nearest neighbors for a candidate selection of similar database descriptors, the execution time scales linearly with both the descriptor dimension and the number of images contained in the database.
   - *Local descriptors* encode an image I_i with a set $D_i = \{d_k | k = 1, ..., K\}$ of vectors $d_k \in \mathbb{R}^d$ at K regions of interest. They often provide better performance than

holistic descriptors, but require computationally expensive methods for local feature matching. Therefore, local descriptors are typically used in a hierarchical pipeline, where first the holistic descriptors are used to retrieve the top-k matches, which are then re-ranked using local descriptor matching.

As the descriptor computation is one of the first steps in a pipeline for VPR, it has a significant impact on the performance of subsequent steps and the overall performance of the VPR system. The algorithm used to obtain the descriptors determine how well the descriptors are suited for a specific task. Moreover, the specific training data of deep-learned descriptors affect the performance in different environments.

3. **Descriptor similarity between two images:** To compare the image descriptors of two images, a measure of similarity or distance must be calculated. This process compares the descriptors d_i and d_j (global) or D_i and D_j (local) of images i and j. Note that similarity s_{ij} and distance $dist_{ij}$ can be related through inversely proportional functions such as

$$s_{ij} = \text{-}dist_{ij} \quad \text{or} \quad s_{ij} = \frac{1}{dist_{ij}}$$

   Holistic descriptors can be compared more efficiently than local descriptors, as they only require simple and computationally efficient metrics like the cosine similarity or the negative Euclidean distance. On the contrary, comparing local descriptors requires more complex and computationally expensive algorithmic approaches, as previously mentioned.

4. **The pairwise similarity matrix**: The pairwise descriptor similarity matrix S contains all calculated similarities s_{ij} between the descriptors of images in the database and query sets. S has dimensions $|DB| \times |Q|$ (supposing a multi-session VPR). Depending on the approach used, S may be dense or sparse. The overall appearance of S is influenced by potentially regular camera's trajectories during acquisition of Q and DB. The pattern of high similarities within S can have a significant impact on the performance of the pipeline.

5. **Matching decisions:** The output of a VPR system is a set of matching decisions $m_{ij} \in M$ with $M \in \mathbb{B}^{|DB| \times |Q|}$ that indicate whether the i-th database image and the j-th query image show the same place or not. Existing techniques for matching range from choosing the best match per query or a simple thresholding of the pairwise descriptor similarities to a geometric verification spatial feasibility, using for example the epipolar constraint.

When ground truth is available, these matching decisions need to be evaluated. One of the most common evaluation metrics is the yet mentioned *recall@k*. For each query image, given the K database images with the K highest similarities s_{ij}, the *recall@K* measures the rate of query images with at least one actually matching database image.

In a deep learning based method, at the training stage, a contrastive or a triplet loss are nowadays usually preferred for the optimization step, over the initially used classification loss.

2.4 Tested methods

In this section I will delve a little bit deeper into two image retrieval methods: NetVLAD (Arandjelović et al. [2016]) and Cosplace (Berton et al. [2022]). The reason is that these are the two methods tested in the numerical experiments, as I will show in chapter 4. The aim is to address in general terms how they work and their differences. The cited publications are recommended if a more detailed and technical discussion is needed.

NetVLAD The authors proposed to learn the descriptor $f_\theta(I)$ of an image I in an end-to-end manner, directly optimized for the task of place recognition. The representation is parametrized with a set of parameters θ, and thus also the euclidean distance depends on the same parameters. The problem is posed as to search for the explicit feature map f_θ which works well under the chosen euclidean distance. In NetVLAD a CNN is cropped at the last convolutional layer, so that it can be viewed as a dense descriptor extractor. Namely, the output of the last convolutional layer is a $H \times W \times D$ map which can be considered as a set of D-dimensional descriptors extracted at $H \times W$ spatial locations. Besides that there is a pooling layer inspired by the Vector of Locally Aggregated Descriptors (VLAD) (Jegou et al. [2010]) that pools extracted descriptors into a fixed image representation and its parameters are learnable via back-propagation. VLAD is a pooling method that captures information about the statistics of local descriptors aggregated over the image. Whereas bag-of-visual-words aggregation keeps counts of visual words, VLAD stores the sum of residuals (difference vector between the descriptor and its corresponding cluster centre) for each visual word. Formally, given N D-dimensional local image descriptors x_i as input, and K cluster centres ("visual words") c_k as VLAD parameters, the output VLAD image representation V is $(K \times D)$-dimensional. This matrix is converted into a vector and, after normalization, used as the image representation. The novelty of NetVLAD lies on the fact that the VLAD layer is differentiable with respect to all its parameters and the input, and so trainable via back-propagation. This is achieved by replacing the VLAD hard assignment of descriptors to a cluster, with a soft assignment to multiple clusters. The result is a powerful image representation trainable end-to-end on the target task.

The loss used for the training is a weakly supervised triplet ranking loss. Being q the query image, $\{p_i^q\}$ a set of potential positives, and $\{n_j^q\}$ a set of definite negatives, the loss assumes the form:

$$L_\theta = \sum_j l\Big(min_i d_\theta^2(q, p_i^q) + m - d_\theta^2(q, n_j^q)\Big)$$

where l is the hinge loss $l(x) = max(x, 0)$, m is a constant parameter giving the margin, and d_θ is the euclidean distance. The weakly supervised ranking loss opens up the possibility of end-to-end learning for other ranking tasks where large amounts of weakly

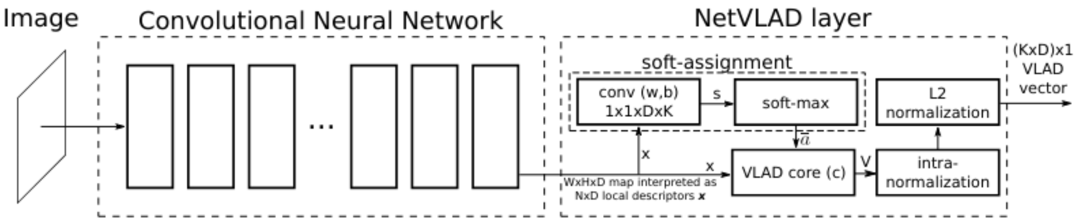


Figure 2.7: CNN architecture with the NetVLAD layer. The layer can be implemented using standard CNN layers (convolutions, softmax, L2-normalization) and one easy-to-implement aggregation layer to perform aggregation. Image from Arandjelovic et al. [2018].

labelled data are available, for example, images described with natural language.

Cosplace The idea behind Cosplace is to treat VPR as a classification task, which is not straightforward given the continuity of the space where the images have been captured. The dataset needs thus to be split into classes, but a naive division into square geographical cells would lead to misclassification of nearly identical images due to quantization errors. To overcome this limitation, the authors proposed not to train the model using all the classes at once, but just groups of nonadjacent classes, called Cosplace groups (see Figure 2.8).

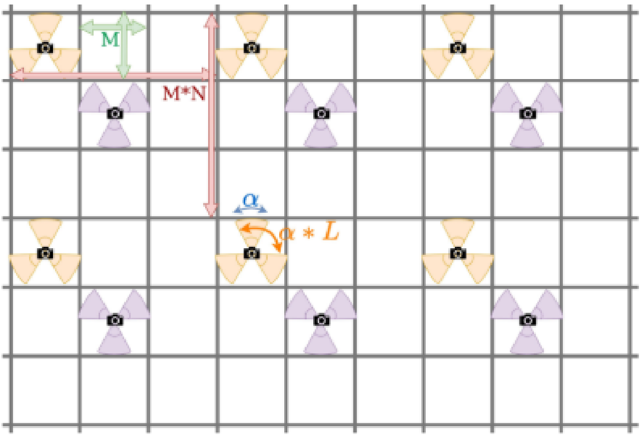


Figure 2.8: Visual representation of Cosplace groups. The orange triangles represent one of the $N \times N \times L$ different groups, while the purple triangles represent another group. Each triangle represents one class, which contains all images within the respective cell with the proper orientation. Image from Berton et al. [2022].

Intuitively, these groups, are akin to separate datasets and the proposed training procedure iterates over them, one at a time. Groups are generated by fixing the minimum spatial separation that two classes of the same group should have, either in terms of translation or orientation. With this partitioning of the dataset, a Large Margin Cosine Loss (LCML)(Wang et al. [2018]) is sequentially performed over each Cosplace group and iterates over the many groups. The remarkable advantage of this procedure with respect to the methods based on contrastive losses, is that no mining nor caching is required, making it a much more scalable option. At validation and test time, the model is used not to classify the query, but rather to extract image descriptors as for a classic retrieval over the database. This allows for the model to be used also on other datasets from unseen geographical areas.

Chapter 3

Structure from Motion

The main subject of this work is Structure from Motion, which is, as already hinted, the problem of recovering the three-dimensional structure of a stationary scene from a set of projective measurements, represented as a collection of two-dimensional images, via estimation of motion of the cameras corresponding to these images. A particularly effective approach to 3D reconstruction involves the use of many images of a stationary scene, that is commonly referred to as *multiview structure from motion*. SfM techniques find applications in many fields, such as robotics, geosciences and culturale heritage maintenance. In this chapter, a brief introduction to the history of this class of algorithms will be followed by a more general and theoretical discussion of their functioning. Finally, replicating the structure of the previous chapter, I will focus on the SfM method used for the numerical experiments: **COLMAP**.

3.1 History of SfM

Longuet-Higgins [1981] introduced the first linear method based on point correspondences, later named the *eight point algorithm*, to solve the SfM problem for a pair of cameras. Specifically, for a pair of cameras, this method aims at estimating the relative camera motion, i.e. relative rotation and translation, and the 3D coordinates of the scene points captured by these cameras, involving a simple pinhole camera model. However Longuet-Higgins does not provide any algorithm for solving this problem. Another early work in the literature that introduced the factorization method for SfM is Tomasi and Kanade [1992]. To model the SfM problem for objects that are relatively distant compared to their sizes, the authors assume an orthographic camera model, in which the 3D points are measured via parallel projections onto the image plane (consequently ignoring the camera translation along the optical axis).

Modern methods usually solve the multi-view SfM problem using bundle adjustment techniques, which aim to optimize a cost function known as the total *reprojection error*. With this cost function, given n images of a stationary scene, the objective is to simultaneously determine the structure (3D coordinates of scene points) and the calibration parameters of each of the n cameras that minimize the discrepancy between image measurements and

their predictive model. The camera location estimation methods is based on corresponding point estimates between pairs of images. In most cases they use the corresponding point estimates to make relative motion measurements between pairs of images, which can be decomposed into pairwise rotational and translational measurements. The generic approach for these methods is to separately estimate the camera orientations based on the pairwise rotational measurements and to use these camera orientation estimates together with the pairwise translational measurements in order to solve for the camera locations. Once the cameras locations and orientations are estimated, these information are used, together with the camera's intrinsic parameters which can be known a priori or estimated as well, to project the points of the images into the three-dimensional space.
Snavely et al. [2006] presented a sequential pipeline for SfM, demonstrating that it can produce accurate reconstructions in practical scenarios where hundreds, or even thousands of independently captured photographs are provided, generating a huge interest in the development of efficient SfM techniques for large, unordered images sets. A notable work in this sense is described in the paper *Building Rome in a Day* (Agarwal et al. [2009]), where the authors pioneered city-scale reconstruction not using images from a structured source, but chaotic and noisy photos from the Internet. One of the current state-of-the-art algorithms is called COLMAP (Schönberger and Frahm [2016]), and will be discussed in more details in 3.3.

3.2 Theoretical background

The feasibility of the task is granted by Ullman's theorem (Ullman [1979a]), from which the name "structure from motion" itself comes from:

Theorem 3.1. *Given three distinct orthographic views of four non-collinear points in a rigid configuration, the structure and motion compatible with the three views are uniquely determined.*

The proof, based on an ideal experiment with two coaxial cylinders is here omitted, but can be found in Ullman [1979b].
I will focus on incremental SfM, a sequential processing pipeline with an iterative reconstruction component commonly used in recent methods. The iterativeness allows to first initialize the stereo view as a two-view geometry. Figure 3.1, from Schönberger and Frahm [2016], shows a generic incremental SfM strategy.

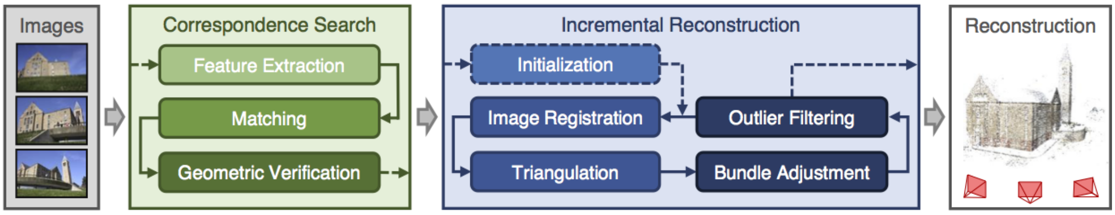


Figure 3.1: A generic incremental Structure from Motion pipeline. Each step will be discussed in detail in the following pages.

Let us represent the i-th camera using its orientation $R_i \in SO(3)$ and its location $t_i \in \mathbb{R}^3$ by (R_i, t_i). Considering that we can always fix the intrinsic coordinate system of one of the two cameras to be the global coordinate system (i.e., we can set $R_i = \mathcal{I}$ and $t_i = (0{,}0{,}0)$, since an absolute coordinate system cannot be determined from point correspondences), solving for the relative motion for a cameras pair, and for the scene points based on the computed motion, then corresponds to actually solving the SfM problem for the pair. A scene point $\boldsymbol{X} = (X^1, X^2, X^3)^T \in \mathbb{R}^3$ is represented in the i-th image plane by $\boldsymbol{x}_i \in \mathbb{R}^3$. Formally, we obtain $\boldsymbol{x}_i$ by first representing $\boldsymbol{X}$ in the i-th camera's coordinate system, as $\boldsymbol{X}_i = R_i^T(\boldsymbol{X} - t_i) = (X_i^1, X_i^2, X_i^3)^T$ and then projecting it to the i-th image plane by

$$\boldsymbol{x}_i = \frac{f_i}{X_i^3}\boldsymbol{X}_i \tag{3.1}$$

where $R_i \in SO(3)$, $t_i \in \mathbb{R}^3$ and $f_i \in \mathbb{R}^+$ denote the orientation, the location of the focal point and the focal length of the i-th camera respectively. Note that, in general, the focal length is defined as the distance between the focal point (here the position of the camera) and the image plane. For a pair of cameras i and j we can restate the coplanarity of the points $\boldsymbol{X}$, t_i and t_j

$$\left((\boldsymbol{X} - t_i) \times (\boldsymbol{X} - t_j)\right)^T (t_i - t_j) = 0 \tag{3.2}$$

in terms of the observable corresponding point measurements $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and the camera parameters as the *epipolar constraint*:

$$\boldsymbol{x}_i^T \left([R_i^T(t_j - t_i)] \times R_i^T R_j \right) \boldsymbol{x}_j = \boldsymbol{x}_i^T E_{ij} \boldsymbol{x}_j = 0 \tag{3.3}$$

where $E_{ij} = R_i^T(t_j - t_i) \times R_i^T R_j$ is the *essential matrix* for the cameras i and j.

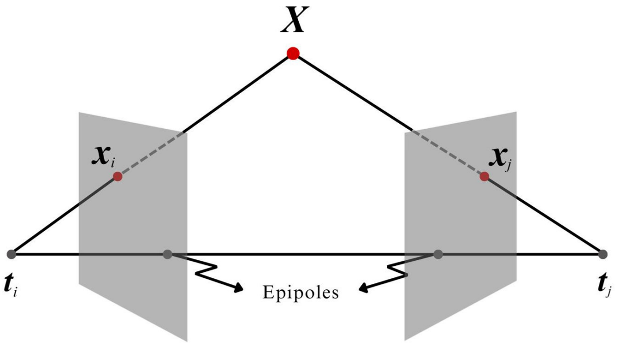


Figure 3.2: Epipolar constraints on two cameras view.

Also let $T_{ij} = [R_i^T(t_j - t_i)]_\times$, where $[M]_\times$ is the skew matrix corresponding to the vector product with M, and $R_{ij} = R_i^T R_j$, yielding $E_{ij} = T_{ij} R_{ij}$. Among the various equivalent restatements of the coplanarity of $\boldsymbol{X}$, t_i and t_j, the useful property of (3.3) is that it

provides a basis for the estimation of the specially structured essential matrices in terms of the observable corresponding points. In other words, by fixing the undetermined scale for the entries of E_{ij} (e.g., $||E_{ij}||_F = 1$ or $||t_j - t_i|| = 1$), we can solve for $E_{ij} \in \mathbb{R}^{3\times 3}$ from eight (linearly independent) epipolar constraints (3.3) corresponding to eight 3D points. The usual approach in the literature is simply to minimize the sum of squared errors in the epipolar constraints subject to the scale constraint, which is equivalent to finding the singular vector of the resulting data matrix corresponding to its smallest singular value. After estimating E_{ij} (up to a sign to be determined later), one can solve for T_{ij} (up to another sign, again, to be determined later) by eliminating R_{ij} using $E_{ij}E_{ij}^T = T_{ij}T_{ij}^T$, and then for R_{ij}, based on its orthogonality, and for the scene points using simple algebraic equations. Lastly, the undetermined signs of E_{ij} and T_{ij} are fixed by requiring the scene points to lie in front of both of the cameras.
The essential matrix estimates, computed for example using the epipolar constraints, in the presence of a sufficient number of corresponding scene points (and the knowledge of the intrinsic camera parameters), can be uniquely factorized into relative rotational and translational parts. In general, the relative rotational parts are used for the estimation of camera orientations R_i. In order to estimate the camera locations, the estimated orientations can be used directly with the translational parts to obtain estimates of the pairwise directions

$$\gamma_{ij} = (t_i - t_j)/||t_i - t_j|| \tag{3.4}$$

A crucial point to note about the estimates of the pairwise directions γ_{ij} is that they lack any scale information pertaining to the distance $||t_i - t_j||$ between camera pairs. The difficulties arising from this homogeneity can be regarded as the most important contributor to the diversity of the camera location estimation methods in the literature. A relatively direct approach for location estimation from pairwise directions is based on the homogeneous system of equations

$$(I - \gamma_{ij}\gamma_{ij}^T)(t_i - t_j) = 0 \tag{3.5}$$

which encapsulates the fact that the difference vectors $t_i - t_j$ are parallel to the pairwise directions γ_{ij}. The locations can be estimated by minimizing the sum of squared errors in the system obtained by replacing γ_{ij} in (3.5) with their estimates and using constraints to fix the global scale and translation, given respectively by $\sum_i ||t_i||^2 = 1$ and $\sum_i t_i = 0$, to prevent the trivial solution of $t_i \equiv t$.

Once the transformation and orientation matrix between adjacent cameras are calculated and each pair of points' 2D coordinates are known, a triangulation method is used to obtain the spatial coordinates of 3D points. A new scene point can be triangulated and added to the point cloud as soon as at least one more image, also covering the new scene part but from a different viewpoint, is registered. Triangulation is a crucial step in SfM, as it increases the stability of the existing model through redundancy, but often these methods suffer from limited robustness or high computational cost. Thereafter, new images in the remaining multiple-view set can be added to the 3D model scene iteratively by solving the Perspective-n-Point (PnP) problem (Figure 3.3).

Since the problem is typically outlier-contamined, robust solvers, such as RANSAC (Lepetit V. [2009]) are usually employed. For uncalibrated cameras, various minimal

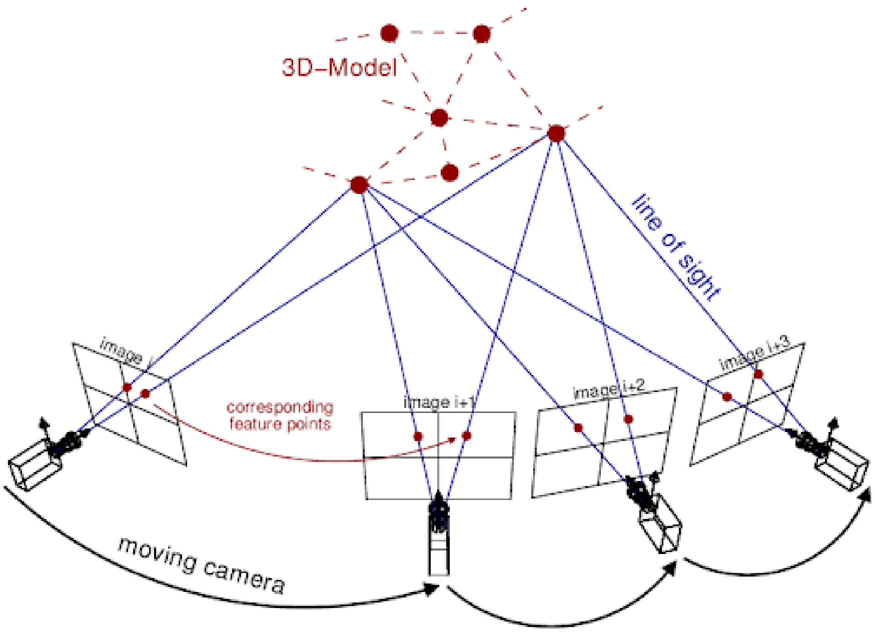


Figure 3.3: PnP problem in reconstruction process.

solvers can be used to estimate the camera intrinsics as well.

Uncertainties in the camera pose propagate to triangulated points and vice versa, and additional triangulations may improve the initial camera pose through increased redundancy. Without further refinement, SfM usually drifts quickly to a non-recoverable state. Bundle Adjustment (BA) (Triggs et al. [2000]) is the joint non-linear refinement of camera parameters $\boldsymbol{C}_i$, which includes both extrinsic parameters like orientation and position and intrinsic ones, and 3D points parameters $\boldsymbol{X}_k$ that minimizes the reprojection error:

$$E = \sum_h \rho_h \Big(||\pi(\boldsymbol{C}_i, \boldsymbol{X}_k) - \boldsymbol{x}_h||_2^2 \Big) \tag{3.6}$$

using a function π that projects scene points to image space and a loss function ρ_h to potentially down-weight outliers and enhance robustness. Levenberg-Marquardt method is a common choice for solving BA problems.

In conclusion let me pose some more attention on the initial stages of the pipeline. It should be clear now that the points selection is crucial, and this suggests that the points should not be sampled randomly over an image, but it should be a representative point, in some sense. For this aim, the first step of every SfM algorithm requires a local features extractor, whose outputs are, as I mentioned in 2, collections of keypoints of the image. The features should be invariant under radiometric and geometric changes so that SfM can uniquely recognize them in multiple images. SIFT (Lowe [1999]), its derivatives (Tuytelaars and Mikolajczyk [2008]), and more recently deep learned features extractors such as

DISK (Tyszkiewicz et al. [2020]) or SuperPoint (DeTone et al. [2018]) are common choices.

Thereafter, SfM discovers images that see the same scene part by leveraging the local features as an appearance description of the images. The naive approach tests every image pair for scene overlap. It searches for keypoints correspondences by finding the most similar features across the two images, using a similarity metric comparing the points appearance. This approach has computational complexity $O(N^2M^2)$, if N is the number of images and M is the number of keypoints extracted per image, and is prohibitive for large image collections. The output is a set of potentially overlapping image pairs and their associated feature correspondences. There exist many matchers, in some cases even deep learned, that try to do this operation in an efficient way. However an upstream possible solution is to input to the matcher only some pre-selected pairs. This task is perfect for a VPR method, which can compare images based on global descriptors instead of local ones, improving efficiency. Usually in this type of approach, each image is paired with its k most similar ones in the database by a retrieval method, where k can be tuned depending on the type and number of images in the collection. Then the pairs are passed to the matcher to compare local features as well. The retrieval step, as I will show with numerical examples in chapter 4, not only avoid useless local features comparisons speeding up the process, but also reduce the number of points erroneously matched, improving the reconstruction performance.

Since matching is based solely on appearance, it is not guaranteed that corresponding features actually map to the same scene point. Therefore, SfM verifies the matches by trying to estimate a transformation that maps feature points between images using projective geometry. If a valid transformation maps a sufficient number of keypoints between the images, they are considered geometrically verified. Since the correspondences from matching are often outlier-contaminated, robust estimation techniques, such as RANSAC, are required. The verified matches and respective keypoints are finally processed by the incremental mapper as described above.

3.3 COLMAP

While many algorithms previous to COLMAP (Schönberger and Frahm [2016]) could handle the diverse and complex distribution of images in large-scale Internet photo collections, they frequently failed to produce fully satisfactory results in terms of completeness and robustness. Oftentimes, the systems failed to register a large fraction of images that empirically should be registrable, or the systems produced broken models due to misregistrations or drift. First, this might be caused by correspondence search producing an incomplete scene graph, second, by the reconstruction stage failing to register images due to missing or inaccurate scene structure, since image registration and triangulation have a symbiotic relationship in which images can only be registered to existing scene structure and scene structure can only be triangulated from registered images.

COLMAP addresses these challenges proposing the following contributions:

- a geometric verification strategy that augments the scene graph with information about the type of transformation connecting two images and detects images with watermarks;
- a next best image selection maximizing the robustness and accuracy of the incremental reconstruction process;
- a robust triangulation method that produces significantly more complete scene structure than the previous state-of-the-art at reduced computational cost;
- an iterative BA strategy, incorporating re-triangulation and outlier filtering that significantly improves completeness and accuracy;
- a more efficient BA parameterization for dense photo collections through redundant view mining.

The general pipeline previously presented outputs what is called a *sparse reconstruction*, where the structure is estimated from a sparse set of features. COLMAP allows to achieve *dense reconstructions* as well, where the structure is estimated from a dense region of pixels. However, even if this second type of reconstruction is usually more aesthetically pleasing, it requires in input calibrated cameras. Hence, when cameras intrinsic parameters are unknown, a sparse reconstruction has to be performed formerly, and the estimated camera intrinsics are used as an input for the dense reconstruction. In this work I will discuss only sparse reconstructions, both in this section and in the numerical experiments. This choice is made mainly for computational time reasons and because sparse methods are sufficient for the aim of this thesis.

Firstly, COLMAP estimates a fundamental matrix, which is similar to the essential matrix, with the difference that the fundamental matrix does not need the camera intrinsics, and is thus used for uncalibrated images. If at least N_F inliers are found, the image pair is considered as geometrically verified. Next, the transformation is classified by determining the number of homography inliers N_H for the same image pair, to distinguish between pure rotation and planar scenes. Furthermore, a frequent problem in Internet photos are watermarks, timestamps, and frames (WTFs) that incorrectly link images of different landmarks. COLMAP detects such image pairs by estimating a similarity transformation with N_S inliers at the image borders. Any image pair with $N_S/N_F > \epsilon_{SF}$ is considered a WTF and not inserted to the scene graph. For valid pairs, it labels the scene graph with the model type (general, panoramic, planar) alongside the inliers of the model with maximum support. The model type is leveraged to seed the reconstruction only from non-panoramic and preferably calibrated image pairs. An already augmented scene graph enables to efficiently find an optimal initialization for a robust reconstruction process. In addition, it does not triangulate from panoramic image pairs to avoid degenerate points and thereby improve robustness of triangulation and subsequent image registrations.

Thereafter, COLMAP propose an efficient next best image strategy following an uncertainty driven approach that maximizes reconstruction robustness. Choosing the next best view is critical, as every decision impacts the remaining reconstruction. A single bad decision may lead to a cascade of camera mis-registrations and faulty triangulations. In addition, choosing the next best view greatly impacts both the quality of pose estimates and the completeness and accuracy of triangulation. A popular strategy is to choose the image that sees most triangulated points with the aim of minimizing the uncertainty in camera resection. For Internet photos, the standard PnP problem is extended to the estimation of intrinsic parameters in the case of missing or inaccurate prior calibration. A large number of 2D-3D correspondences provides this estimation with redundancy, while a uniform distribution of points avoids bad configurations and enables reliable estimation of intrinsics. The used approach approximates the intuition of Haner and Heyden [2012] of an uncertainty driven approach using an efficient multi-resolution analysis. It simultaneously keeps track of the number of visible points and their distribution in each candidate image. More visible points and a more uniform distribution of these points should result in a higher score, such that images with a better-conditioned configuration of visible points are registered first.

Especially for sparsely matched image collections, exploiting transitive correspondences boosts triangulation completeness and accuracy, and hence improves subsequent image registrations. Approximate matching techniques usually favor image pairs similar in appearance, and as a result two-view correspondences often stem from image pairs with a small baseline. Leveraging transitivity establishes correspondences between images with larger baselines and thus enables more accurate triangulation. Hence, feature tracks are formed by concatenating two-view correspondences. Unfortunately feature tracks often contain a large number of outliers due to erroneous two-view verification of ambiguous matches along the epipolar line. A single mismatch merges the tracks of two or more independent points. Hence COLMAP's authors propose a recursive triangulation scheme to recover the potentially multiple points of a feature track from a faulty merge, combined with the formulation of the problem of multi-view triangulation using RANSAC, to handle arbitrary levels of outlier contamination.

To mitigate accumulated errors, BA is performed after image registration and triangulation. Usually, there is no need to perform global BA after each step, since incremental SfM only affects the model locally. Hence, local BA is performed on the set of most-connected images after each image registration. Global BA is executed only after growing the model by a certain percentage, resulting in an amortized linear run-time of SfM. To account for potential outliers, COLMAP employs the Cauchy function as the robust loss function ρ_j in local BA. For problems up to a few hundred cameras, a sparse direct solver is used, and for larger problems PCG. For unordered Internet photos, it relies on a simple camera model with one radial distortion parameter, as the estimation relies on pure self-calibration. After BA, observations with large reprojection errors are filtered. After global BA, degenerate cameras are also filtered, e.g. those caused by panoramas or artificially enhanced images. re-triangulation is performed both before and after global BA, to account for drifts effects and improve the completeness of the reconstruction. BA, re-triangulation, and filtering are executed in an iterative optimization until the number of filtered observations and post-BA re-triangulation points diminishes. In most cases, after the second iteration results improve dramatically and the optimization converges.

BA is a major performance bottleneck in SfM. Hence COLMAP lastly proposes a method that exploits the proper characteristics of incremental SfM and dense photo collections for a more efficient parameterization of BA by clustering redundant cameras into groups of high scene overlap. Internet photo collections usually have a highly non-uniform visibility pattern due to varying popularity of points of interest. Moreover, unordered collections are usually clustered into fragments of points that are covisible in many images. The proposed approach exploits this facts to improve the efficiency of BA. The problem is partitioned into submaps, whose internal parameters are factored out. The scene is partitioned into many small, highly overlapping camera groups. The cameras within each group are collapsed into a single camera, thereby reducing the cost of solving the reduced camera system. I will not go any further on this contribution, since it prescinds the focus of this thesis, but technical details can be found in Schönberger and Frahm [2016].

Chapter 4

Conducted experiments

In this chapter I will investigate the importance of image retrieval in a SfM pipeline by means of numerical experiments. In particular, in section 4.1 I will briefly talk about some of the early tests on some various image sets. These experiments do not represent the main focus of the thesis, but they paved the way for the setting of the main experimental scenario. In section 4.2 I will focus on the image sets choice. In section 4.3 I will discuss the problem of evaluating a 3D reconstruction and introduce the proposed evaluation strategy. Finally, in section 4.4 the numerical experiments and their results will be reported and discussed.

4.1 Early Tests

Conducting the first experiments I was not aware of what would have been the point of this thesis. I knew I wanted to study the role of image retrieval in a 3D reconstruction scenario, but I did not know exactly what turn this work would take. This premise explains why these tests are quite disconnected one from another, to explore some various scenarios to understand where to focus later. Although the results are not very interesting from a research point of view, I believe they have been useful as a first approach to this field and guided me to the setting of the experiments that will be discussed in next sections.

The basic idea was to apply a SfM pipeline, concerning an image retrieval step, on some scenes and evaluate the reconstruction obtained with COLMAP, a state-of-the-art method described in section 3.3. Since the focus was directed towards image retrieval, in the sense of matching similar images to enhance the performance of an SfM pipeline, I first posed my attention on 2023 Image Matching Challenge (Chow et al. [2023]). This is a competition where participants have to reconstruct a 3D model of some scenes. The 2023 edition however was mainly based on scenes of limited size. Most of the solutions proposed by participants either did not use image retrieval, or tested it but stated it was useless. I wandered why and I conducted some tests on the 7 scenes provided as training sets. The problem was that most of the sets were a small collection of images depicting a single object of modest scale, e.g. a bike or a fountain. The only scene that was larger in the number of images contained 174 images of the ruins of a temple, located in Sicily. Here the issue was that many images depicted small particulars of the buildings, making the retrieval method match pictures of different areas but looking globally similar. The last scene contained instead photos taken around a building, the Kyiv Puppet Theatre. Here the images were more varied, but given the small cardinality of the set (only 27 images), VPR yield no big enhancement to the reconstruction, neither in terms of accuracy nor of time.

Consequently I moved to a dataset for which image retrieval could have more weight. The choice went on the new edition of the same challenge. In 2024 edition (Bellavia et al. [2024]) the goal was the same as the previous year, but the image scene provided were more varied. However in this case the focus was set on pathological image matching cases. As an example, some scene contained small sets of glass objects, others strongly symmetric and repetitive buildings, others as well low resolution images of a park. The result is that, after some tests, image retrieval showed up to be, again, quite useless.

This made me think that for this type of investigation, I needed a large number of images, possibly covering a quite large scene. In this scenario, many images would not share a covisibility area, and would be wrongly matched without a first matching based on global descriptors. I moved to a well known dataset in computer vision, named *San Francisco XL* (Berton et al. [2022]). This is a large scale dateset that covers the whole city of San Francisco with images taken from Google Street View between 2009 and 2021. In particular, the authors collected 3.41 millions of 360° panoramas and extracted 12 horizontal crops for each. The total number of images available for training amounts to 41.2 millions, all paired with GPS and heading labels. Obviously I made tests on a subset of this very large set, choosing a 10000 m^2 square randomly picked on the map of San Francisco. This resulted in 1375 images. The first reconstruction obtained were highly confused and a structure of the street was hard to identify. The problem, this time, was the source of the photos. As previously stated, these images are crops of 360° panoramas. This gave them an intrinsic distortion that, even if it could seem a secondary aspect from a human perspective, made it almost impossible to perform the reconstruction and obtain a satisfying result. Indeed, the distortion changes the distance between points in the image space, causing issues to the geometric verification, triangulation and pose estimation discussed in 3.2. Fortunately, knowing the camera intrinsic parameters, it was no big deal to undistort the images, as shown in the example in Figure 4.1.

The new reconstruction, with undistorted images, finally gave appreciable results. The 1375 images, all registered into the 3D model, returned the recognisable shape of the street, and the estimated camera poses looked compatible with the Google Street View source of the images, as shown in Figure 4.2.

Different retrieval approaches, anyway, resulted in similar reconstructions, hence I tried to enlarge the scene, to make the image matching more relevant.

The new experiments, on a 1000000 m^2 square were unsatisfactory. The high number of images, above 360000, made unfeasible a systematic experimentation on the computation devices available, where a single test took more than 10 days. In addition the images were

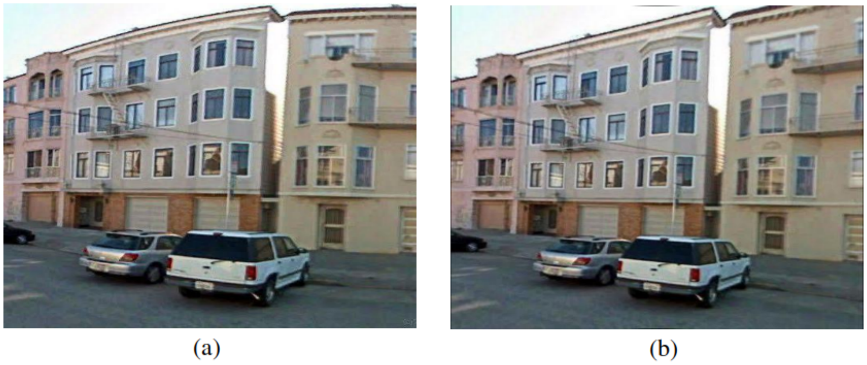


Figure 4.1: Example of a distorted image coming from the cropping of a 360° view from Google Street-View car in (a). The picture is from SF-XL dataset. The result of the applied undistortion is shown in (b).

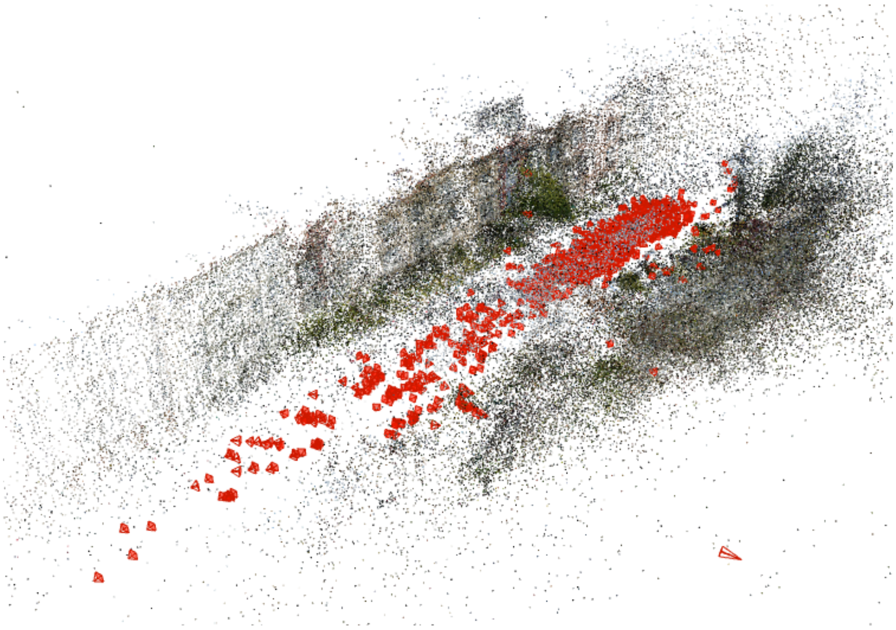


Figure 4.2: One of the reconstructions obtained on the 10000 m^2 square on SF-XL dataset. The red rectangles are the cameras pose estimations, mostly compatible with the Google Street View source of the images.

heavily redundant, since the set contained pictures of the same places from many different years. A possible solution could be a downsampling, but the images were not uniformly distributed over the map. This means that taking, for example, only the images from a single year, covered only a little section of the 1000000 m^2 square.

The same issue appeared with a random sampling: same areas were covered with redundant images while others were not covered at all. I achieved an almost full coverage

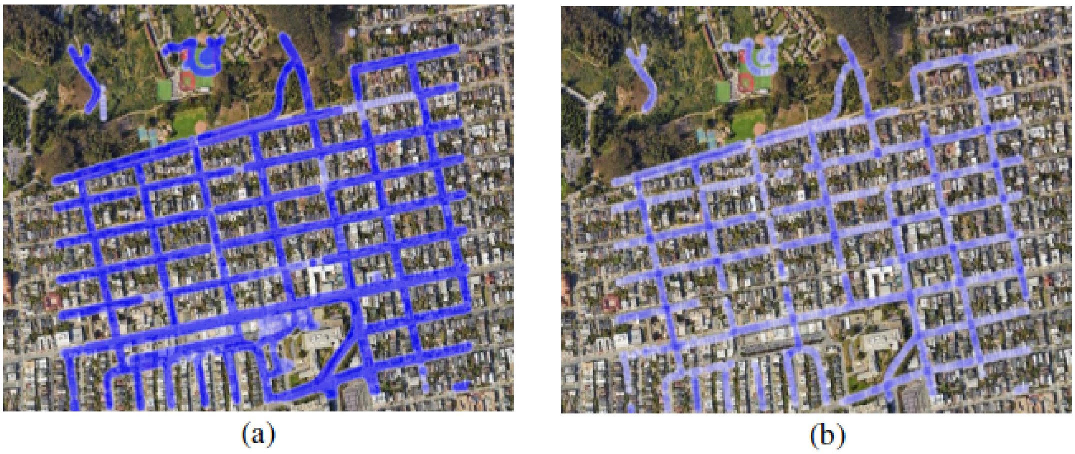


Figure 4.3: Image coverage on the 1000000 m^2 square selected from SF-XL dataset. (a) shows the positions of all the images in the dataset. (b) shows only the images taken in 2009, which is the year with the higher number of pictures. Even if it contains more than 20000 images, there are some uncovered spots, that represent a challenge for the reconstruction process.

selecting the images from the three years with higher cardinality, resulting in approximately 60000 images. Beyond the still long time needed, the result was very poor and unsatisfactory. Different streets were overlapping in the 3D model and reconstructed at different planes on the vertical axis. Probably the fact that all the images were taken from the same point of view (the Google Street View car) did not help the SfM algorithm, that in most cases could only reconstruct the low part of a building facade. One thing I noted right away, is that the values of the only evaluation metric provided by COLMAP (the mean reprojection error, yet presented in 3.3) were quite low, even better than in the experiments on the smaller square. This was clearly in contrast with the awful reconstructions obtained. This suggested that the mean reprojection error is not a reliable and informative metric for the type of experimentation I needed. I will discuss better this issue and my proposed solution in 4.3. Thereafter I decided to abandon this dataset and move to something more aligned with the aim of the thesis, as I will describe in 4.2. These tests led me to seek a large-scale scene, with images from different points of view, an high covisibility but not too much redundancy.

4.2 The Datasets

The scenes chosen for the experiments are made of pictures taken from `Flickr.com`. Flickr is an image and video hosting service, as well as an online community, founded in Canada in 2004. For uploading images only a free account is needed, and this results in the fact that a lot of images can be found on the platform, both professional and amateur. Especially in touristic sites, the amount of photos is quite large, and their variety in terms of cameras, points of view, editing and environmental conditions, makes them well suited for the aim of this thesis. On the other hand, it must be remembered that these data represent a big challenge for the SfM task. Hence, it should be no wonder if the reconstructions will not be as good as other ones that can be found, maybe relying on more structured inputs. The same tests conducted could be obviously done on indoor scenes as well, but I used only outdoor ones for larger image availability reasons.

Flickr images can be easily downloaded with an API. Selecting pictures by the search of one or more words in the image titles is not a good choice, since the titles often do not depict the real content of the image and this would drastically reduce the pool of available photos. A search on the tags can give better results. Anyway not all users use tags on their pictures, and this strategy can mainly be used to find photos of famous landmarks. I did not want to have this constraint and at the same time to get too many images that would make unfeasible a wide testing, so I took an alternative path. Some images on Flickr are uploaded together with some metadata, among which there could be the geographical coordinates of where the photo has been taken. This is often done automatically if the camera used is the one of a smartphone, thus this information can be found on many pictures. My approach consists in setting four coordinates, which are respectively minimum and maximum latitude, and minimum and maximum longitude, and downloading all the images taken in the rectangle area within these bounds.

To choose the scenes, I oriented myself towards squares rather than single buildings or monuments, to have a 360° view to reconstruct, and increase further the benefits from the use of image retrieval. I did not choose too extended scenes, such as a district, since in these cases usually many sub-models are reconstructed and then merged together, but for this work this would be a useless effort. I looked for distinctive scenes, to make a first evaluation “by eye” easier, but not too famous ones, to keep the number of images limited, and so the time needed for a reconstruction.

I selected four scenes: three for wide testing and one for a particular experiment that will be better introduced later. The following are the chosen sets:

- **Torino - Piazza Castello**: in this scene I selected a slightly larger area to include the totality of the square, but, as an adverse effect, also some of the adjacent streets. The square has an irregular shape, with a tower on one side and a building, with two round towers, in the centre. It is mainly pedestrian area, but there are tramway lines going through it. Many of the pictures turned out to not be actually depicting the square, thus more than a half of them will be discarded in almost all the reconstructions. The set contains 11752 images (Figure 4.4).
- **Barcelona - Plaça d’Espanya**: this square has a round shape, so some surroundings spots have to be taken necessarily since I am selecting a rectangular area. It is essentially a big roundabout with a fountain in the centre. On the sides, notable buildings are the two Venetian towers, the bullring and a conspicuous building at the end of a street starting from the square. Also in this case many images will

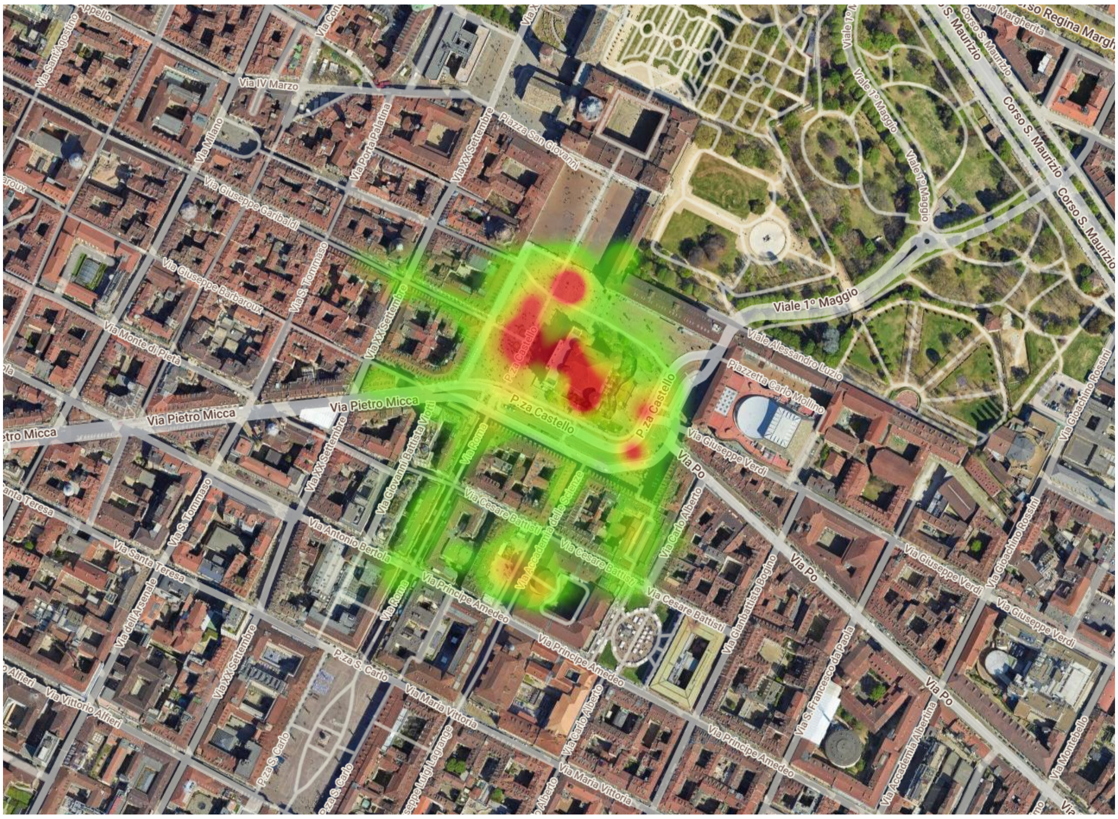


Figure 4.4: Heatmap of images' locations in the Turin scene.

be discarded, for example because many of them have been taken inside one of the museums in that area. The set has a similar size to the previous one, as it contains 11577 pictures (Figure 4.5).

- **Venezia - Piazza San Marco**: this square has an almost rectangular shape. Notable elements are the San Marco bell-tower and the basilica. It is a pedestrian area, but the large number of tourists present in the majority of the pictures, make image matching a tough job. I wanted to conduct some experiments on scene of a different scale, so I randomly sampled 4100 images, out of the almost 15000 found in the area (Figure 4.6).

- **New York - Statue of Liberty**: this scene will not be used in many experiment as the previous ones, but only in one comparison. I wanted to prove numerically the usefulness of image retrieval in a SfM pipeline. As I already told in the introduction, image retrieval allows both to reduce reconstruction time and increase accuracy of the model. The first point is quite obvious, since it reduces the number of image pairs to be geometrically matched, but the second one is a bit more counter-intuitive. Hence I wanted to conduct a test without using image retrieval, thus pairing each picture exhaustively with the others, and compare the result with another experiment

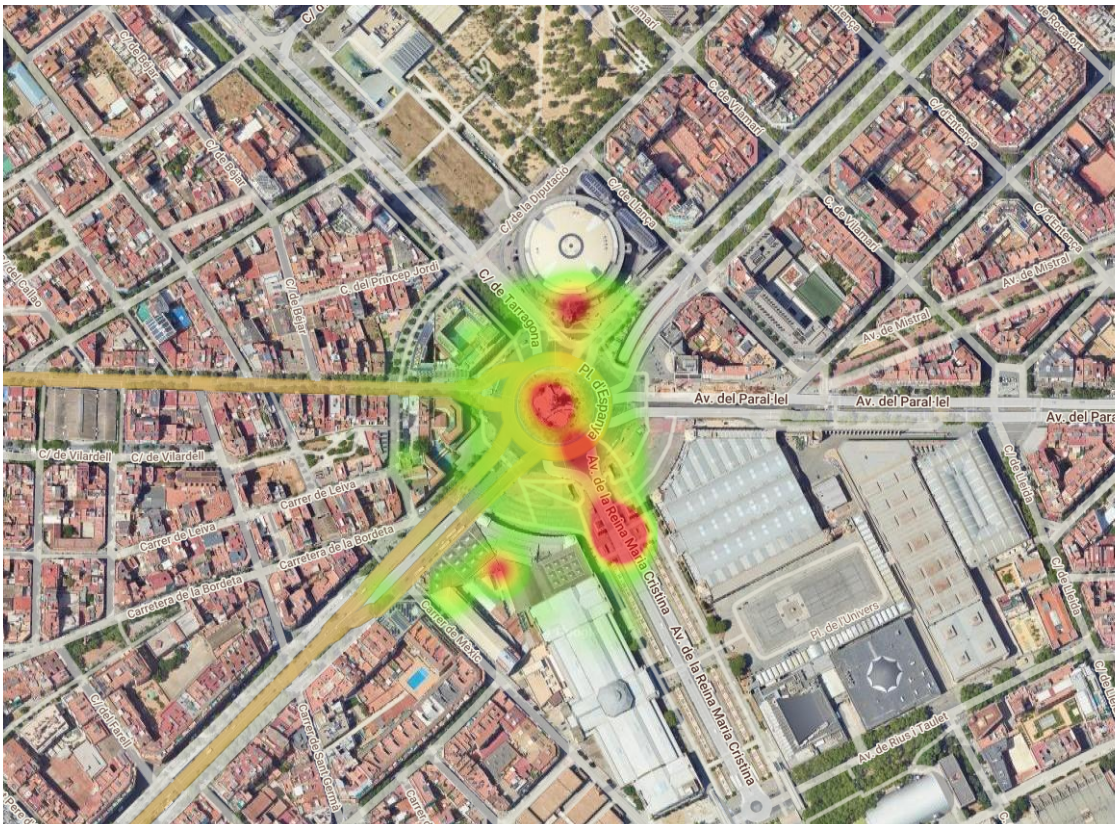


Figure 4.5: Heatmap of images' locations in the Barcelona scene.

exploiting VPR. To do this I needed a very small scene, so that the reconstruction without retrieval would still be performed in an acceptable time. After all, it is not the aspect of a model that I am interested in, but the comparison between different strategies. For this test I selected the approximately 7000 images taken on Liberty Island and I sampled 500 of them. In this case I did not choose a square but a single monument because, with such a limited number of photos, a small subject will probably be easier to reconstruct.

The main issues with this datasets are due to the poor accuracy of the geo-tagging in some images, whose consequence is that many images of different spots of the cities, were wrongly localized in those squares. In addition, sometimes the geo-tagging is not automatically added by the smartphone, but is inserted manually by the user. In this cases, an image can be consciously located in the wrong place. Another problem are the indoor images, maybe taken in some building of the selected area, but clearly useless for the reconstruction. Then, many photos are taken in particular moments, such as events, festivals, protests, or simply with one or few people in the foreground, and a out-of-focus background. These pictures are obviously hard to match to other ones and may lead to mistakes. Lastly, as already hinted, a rectangular selection region not always allows to

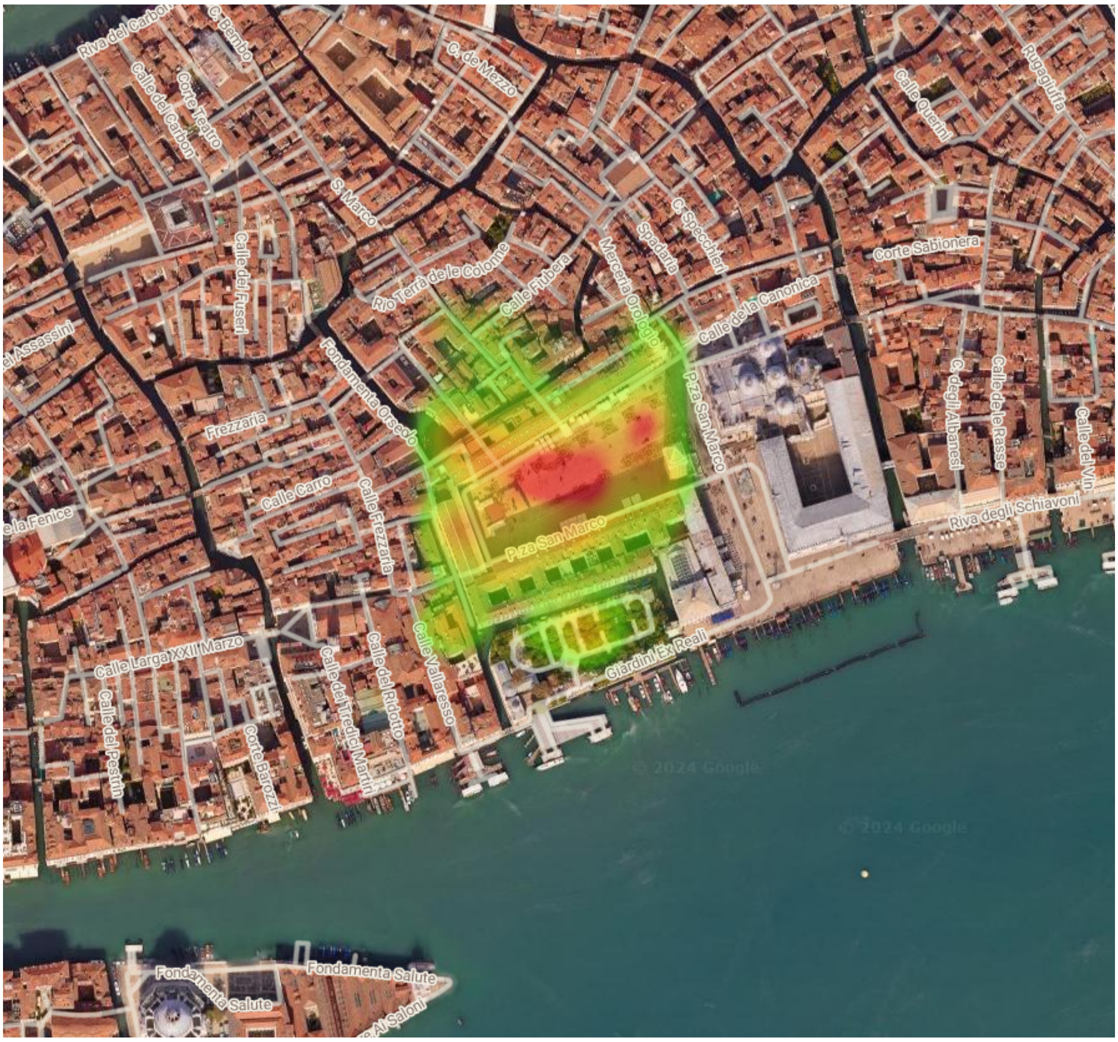


Figure 4.6: Heatmap of images' locations in the Venice scene.

cover precisely the area of the square, introducing images of different places taken in the surrounding streets or buildings.

4.3 The Evaluation Method

As hinted in the previous section, one of the biggest challenges has been finding a way to evaluate a 3D reconstruction. Sometimes it is easy to say if a reconstruction is better or worse than another one by eye, but to make a valuable comparison between different methods and setting, an objective evaluation strategy is needed. The COLMAP implementation gives, as an output, some information, such as the number of images registered, the number of points in the 3D point cloud and the mean track length. The only evaluation metric is the yet cited *mean reprojection error*, that is used during the bundle adjustment as well. This measure is the mean of the reprojection error of each point in the reconstruction, intended as the euclidean distance between the projection of the point over the 2D image and the corresponding keypoint originally extracted. This metric is expressed in pixels. This strategy has two main issues:

- Firstly it strongly depends on the images registered in the reconstruction. Thus it is fair to use this metric for the optimization step, during the bundle adjustment. But it is not an optimal choice if the aim is to compare two or more different reconstructions. Indeed the models may have registered different images, with different keypoints, and the metric, even if it is a mean, is affected by this variation. Similarly also the use of a different local descriptor extractor may lead to different keypoints. Particularly important is the number of points in the model because, if it has few points, the high reprojection error computed over the outlier has a big influence on the mean reprojection error. On the other hand, a model with the same number of outlier but more points, will have a lower error.

- Secondly, since this metric is expressed in pixels, it is also affected by the size of the images. In a dataset of images all with the same resolution it could work fairly, but if the dataset is strongly heterogeneous, a large distance on a picture with very low resolution, would give erroneously a small error in terms of pixels. And, again, if the set of images changes, the mean reprojection error would not be a meaningful measure anymore.

As a result, in the early tests, some clearly wrong reconstructions had a lower error than discrete ones. Hence I needed an other metric, allowing to compare different reconstructions in a consistent way.

A measure that is often used to evaluate sparse reconstruction is the error of the camera pose estimates, in terms of position and orientation. In the 2023 Image Matching Challenge for example, the evaluation metric is a *mean average accuracy*, that involves computing the relative error with respect to the ground truth of the estimated translation vector and rotation matrix. Then each of this error pairs is thresholded over ten pairs of thresholds. The percentage of accurate samples is computed at every thresholding level, and the results are averaged over all thresholds. This strategy, and similar ones, are certainly effective, but they require the ground truth of the camera poses. Thus they can be used only on strongly structured data, essentially built specifically for this task. The approach that I will now present, on the other hand, can be used on every set of images.

The problem with the mean reprojection error is his behaviour when images and keypoints change. The idea is thus to have a fixed test set on which to calculate the reprojection error. The test set images are initially not passed to COLMAP to perform the reconstruction to evaluate. When the reconstruction is complete these images can be added to the existing reconstruction by means of the COLMAP function *image_registrator*. This is necessary because registering all the images at the same time, the positions and intrinsic parameters of the set images would be affected by the position of the other images, for the own nature on bundle adjustment phase. Adding them at a second time instead allows to have genuine estimates, based only on the points already registered in the reconstruction. COLMAP usually suggest to perform a bundle adjustment using the dedicated command after registering new images, to refine their position. However in this case this is not executed because having accurate positions of the test images is not the point of the experiments.

COLMAP, to run a reconstruction, creates a SQL database, containing all the images, keypoints and matches. The function *image_registrator* takes as an input the database and an existing 3D model and tries to add to the reconstruction all the images in the database not already registered. Actually not all the images are registered, since some of them may not have enough matches with images in the model. Just adding the test images to the database is not enough, since COLMAP would also register some of the images discarded in the first reconstruction, but not part of the test set. And, again, the set of registered images would change from a test to another, making all the efforts futile. The database has to be previously cleaned up from all the images not registered in the first reconstruction, and all the keypoints and matches related to them. This way the only pictures COLMAP will try to register in the second phase are the ones in the test set. One problem could be that not all the images in the test set are registered, especially when working with very heterogeneous and noisy datasets. Nevertheless, this is not a big deal. Indeed the number of registered images in different reconstructions is similar. Usually the discarded images are the worst ones, which are discarded from all the reconstructions. Few variations in the set of registered images are mitigated choosing a sufficiently big test set, so that the error on a single point has little influence on the global mean. The set at the same time must not be too large with respect to the total number of pictures, or many of the test images would not share enough matches and would not be registered. For the used scenes I chose a test set of 100 images.

After this all the points in the 3D model seen by at least one of the test images, are reprojected on the corresponding 2D test image. The distance between the original keypoint and its new reprojection is computed with the euclidean metric. The overall error measure is the mean of all these distances.

The last point to discuss is how to reproject a point from the point cloud to an image. The COLMAP's *incremental_mapper* does this automatically to compute the mean reprojection error but, wanting to consider only the points seen by test images, this has to be done manually. One of the output files of the reconstruction process, *images.txt*, contains for each image all the keypoints, and in particular their 2D coordinates and their distinctive *id* in the point cloud. For each image also the position vector, the quaternion giving the orientation and the *id* of the camera associated to the picture can be found in this file. Filtering to only the images of the test set, the 3D coordinates of the points registered in the model can be found, in the other file *points3D.txt*, together with some other information not useful for my objective. For the point projection the camera intrinsic parameters are also needed. They can be found in the third output file *cameras.txt*.

COLMAP supports 11 different camera models. The one suggested when the intrinsics are unknown and every image has a different camera calibration, e.g., in the case of Internet photos, is the *simple radial*. This only models radial distortion, shown in Figure 4.7, with one parameter.

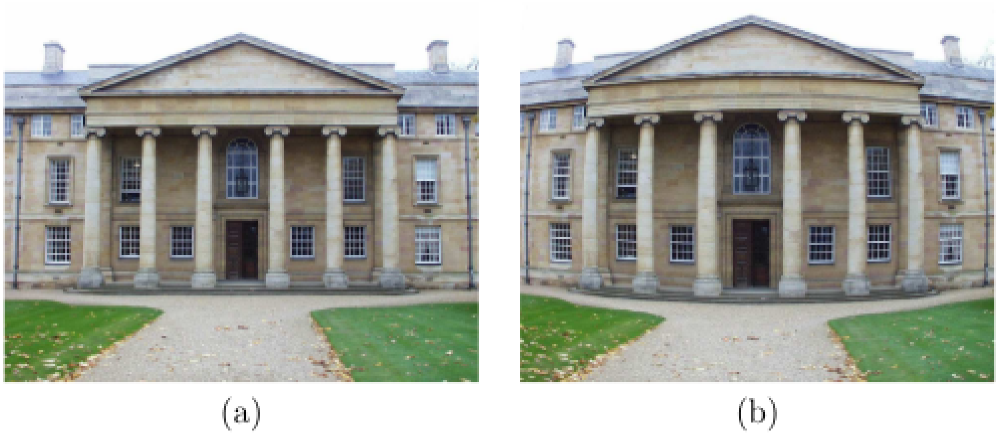


Figure 4.7: Example of radial distortion in image (b). Image (a) is the same image but undistorted.

The other parameters are the *focal length*, already introduced in section 3.2, and the *principal point*, i.e. the intersection of the optical axis and the image plane. An even simpler model is the *simple pinhole* model (Figure 4.8). This is similar to the previous one but it does not take into account a possible distortion.

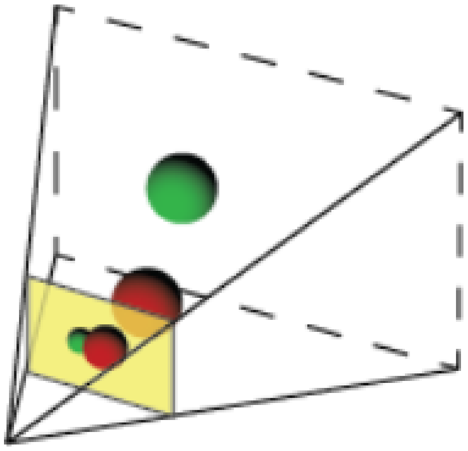


Figure 4.8: Example of the simple pinhole camera model. The used simple radial model is similar to this, but in addition it takes into account a possible radial distortion.

I made the decision to use the *simple radial* model even if, looking at the images taken from flickr, very few of them present a distortion. As intuitive, the reprojection error on the strongly distorted images, such as fisheye ones, is very high. However this is not a big deal because first of all these images are statistically a narrow subset, as already told, and then because the problem occurs in every reconstruction. A distorted image will either not be registered, and so it will not influence the error, or it will have an high error in all the reconstructions, raising the average error but not compromising the comparison of models. Anyway, considering a radial distortion parameter does not solve this problem, since wide-angle lenses would need more complex models, requiring the knowledge of camera intrinsics. Some of the images from flickr have these information in their metadata, but only a little portion of them. Thus I preferred to consider a general case in which nothing about the camera is known a priori.

Focusing on the *simple radial* camera, its intrinsic matrix is parameterized by Hartley and Zisserman [2004] as:

$$K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Here f_x, f_y is the focal length, defined as the distance between the pinhole and the image plane, measured in pixels. In a true pinhole camera it should be $f_x = f_y$, but in some pathological cases or with calibration errors, they can assume different values. In this case, the resulting image would have non-square pixels. Since COLMAP *simple radial* camera model estimates only one focal length value, I will assume $f_x = f_y = f$. The camera's principal axis is the line perpendicular to the image plane that passes through the pinhole. Its itersection with the image plane is referred to as the principal point, whose coordinates with respect to the image plane origin, conventionally chosen, are x_0 and y_0. The axis skew, s, causes shear distortion in the projected image. The result of a non null value of s, would be parallelogram-shaped pixels, with non-right angles. Since every parameter is estimated, and thus brings some error in itself, I will consider the simple case in which $s = 0$. I made this decision also to reduce the time needed for the reconstruction, and it is justified by the fact that, in some tests in which I estimated the skew parameter too, it was very close to 0 in almost all the images. Finally, also in this case, this simplification may affect a little bit the error computation, but this will occur in every tested model since I am using the same set of images. So the performance comparison would still be meaningful, at least on the same scene. The information in the intrinsic matrix is completed by the extrinsic matrix $[R|t]$, which is the concatenation of the rotation matrix and the translation vector. The projection matrix $\mathcal{M} \in \mathbb{R}^{3\times4}$ is given by the product $\mathcal{M} = K[R|t]$

From a theoretical point of view, let us refer to Figure 4.9 for the notation. The camera center is located in O, the image plane is at a distance f (focal length) from O towards the Z axis. The projection of a point $P = (x, y, z)$ onto the image plane is $P' = (x', y', f)$. The goal is to find the coordinates of P'. For the sake of simplicity I will firstly consider this simple case where the extrinsic matrix is useless. Indeed, with the origin and axes positioning of Figure 4.9:

$$R = \mathcal{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad t = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

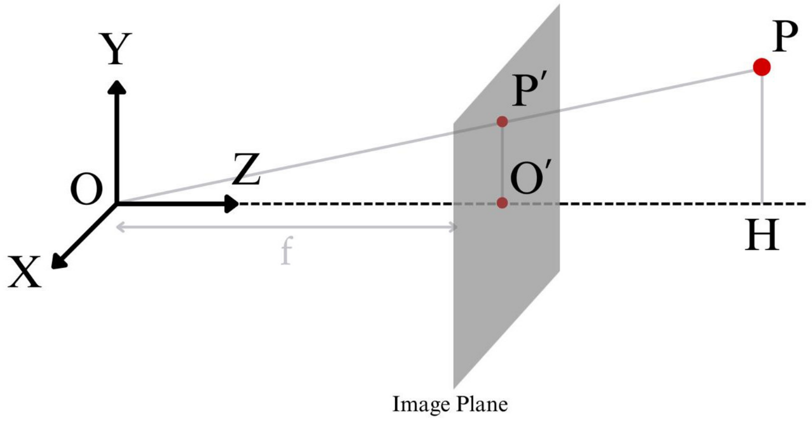


Figure 4.9: Example of point projection setting.

The fundamental observation is that the triangles OHP and $OO'P'$ are similar triangles. Thus, $x'/x = y'/y = f/z$, from which

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x\dfrac{f}{z} \\ y\dfrac{f}{z} \end{pmatrix}$$

These are coordinates in the reference system of the "3D world" $X - Y$ axes. To express them in the reference system of the image plane, the principal point coordinates x_0 and y_0 must be summed to the previous expressions of x' and y' respectively, giving:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x\dfrac{f}{z} + x_0 \\ y\dfrac{f}{z} + y_0 \end{pmatrix}$$

This system can be expressed as a matrix multiplication:

$$\begin{pmatrix} x' \\ y' \\ w \end{pmatrix} = \mathcal{M} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

where the 1 is added in P to have homogeneous coordinates, and $\mathcal{M}$ has this simple form because we are in the trivial case of R and t and we are assuming $s = 0$. The same logic can be extended to the case of a generic extrinsic matrix.
Now, to convert from homogeneous coordinates to euclidean ones, it is sufficient to divide by the last element and then discard it:

$$\begin{pmatrix} x' \\ y' \\ w \end{pmatrix} = \begin{pmatrix} \frac{x'}{w} \\ \frac{y'}{w} \\ 1 \end{pmatrix} \cong \begin{pmatrix} \frac{x'}{w} \\ \frac{y'}{w} \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}$$

Hence, summing up, in the specific experimental setting, it is sufficient to construct the intrinsic matrix K with the parameters estimated by COLMAP and the extrinsinc matrix concatenating the translation vector to the rotation matrix, obtained with simple calculations from the quaternions. The result is the projection matrix $\mathcal{M}$. This matrix is then multiplied to a vector containing the point 3D coordinates and a forth unit coordinate. The first and second values of the resulting vector are divided by the third one, and they are then the coordinates of the point reprojection on the image plane. As already told, the distance of this projection from the keypoint originally extracted is calculated and averaged over all the points seen by test images.

The obtained metric seems to be quite reliable, in the sense that a visibly good reconstruction, usually has a lower error than a worse one, regardless of the number of images in the model. However, it must be remembered that there is some randomness in the reconstruction evaluation, depending on the order in which images are registered. Thus, a difference of few thousandths or even hundredths of pixel must not be considered significant.

4.4 The Numerical Results

First of all let us go through the pipeline used in the following experiments:

1. As already discussed the images are downloaded from Flickr, being selected based on their geo-localization. A test set of 100 photos is randomly sampled and set aside. I will refer to the rest of the images as "train set" for simplicity, even if there is no proper training to be executed on them.

2. The train images are processed for the local feature extraction. COLMAP, by default, uses SIFT (Lowe [1999]) as an extractor, and pretrained weights vocabularies for the matching. Since I wanted to have the possibility to choose different extractors, I relied on the tools provided by `hloc` (Sarlin et al. [2019]), a toolbox for image hierarchical localization, compatible with the python implementation of COLMAP: `pycolmap`. In the majority of tests, the local features extraction has been entrusted to **SuperPoint** (DeTone et al. [2018]).

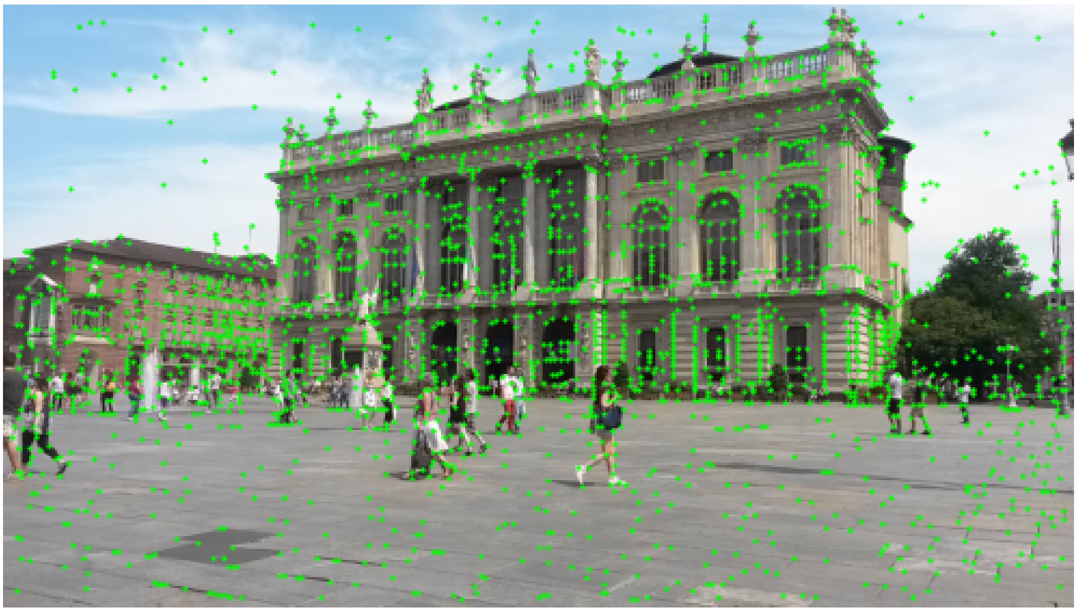


Figure 4.10: Example of keypoints extracted by SuperPoint. The picture is from Turin scene train set.

A convolutional neural network (CNN) is used to produce a keypoint heatmap, highlighting the locations of keypoints with high confidence. Once the keypoints are detected, the descriptor extraction component generates a descriptor for each keypoint. The descriptors are high-dimensional vectors encoding the local appearance around each keypoint. SuperPoint employs a self-supervised learning paradigm. It leverages a synthetic dataset generated through a homographic adaptation process to train the model. This process involves applying random homographies to images to create synthetic correspondences, which are used as supervision signals during training. The joint optimization of keypoint detection and descriptor extraction leads to high-accuracy feature matching, together with efficiency and robustness to changes in viewpoint, illumination, and other variations. The used version of SuperPoint has been previously trained on the dataset *Aachen Day-Night* (Sattler et al. [2018]), making it particularly suitable for outdoor urban scenes. Some tests, only on the Venice dataset, have been carried out with features extracted with SIFT, to see if the results are robust to keypoints changes. SIFT is a quite old, but still well performing, algorithm which focuses on extracting keypoints that are discriminative and both scale and rotation invariant.

3. For the retrieval phase, global descriptors are extracted from the train images, and saved in a database. As already told, the extractors tested in the experiments are NetVLAD and Cosplace. Thereafter for each image the similarity with all the others is computed, based on the descriptors just extracted. The top matches are saved, as

a pair, in a `txt` file. The parameters that can be set for this pair selection are:

- N: the number of images to pair. For example, if $N = 20$, each image is paired with the 20 most similar ones in the dataset. Obviously, self-matching and mutual-matching are handled and properly discarded.
- k: the number of top matches to discard. For example, if $N = 20$ and $k = 5$, not the top 20 matches are selected, but the ones from 6-th to 25-th position, in a list ranked on the similarity score.
- *min_score*: is the minimum similarity score to be accepted. If some of the top N results have a score (which should be a number between 0 and 1) lower than this value, these pairs are discarded.

4. For each pair selected, the local keypoints of the two images are then matched. The matching is entrusted, in the majority of tests, to **SuperGlue** (Sarlin et al. [2020]), a robust method that takes advantage of a graph neural network to learn the matching process directly from data, and to consider the context of all keypoints in an image pair simultaneously. An attention mechanism helps the network focus on the most relevant keypoints and their context, improving the robustness of the matches. For the experiments using SIFT, the matching relies on a simple Nearest Neighbours algorithm.

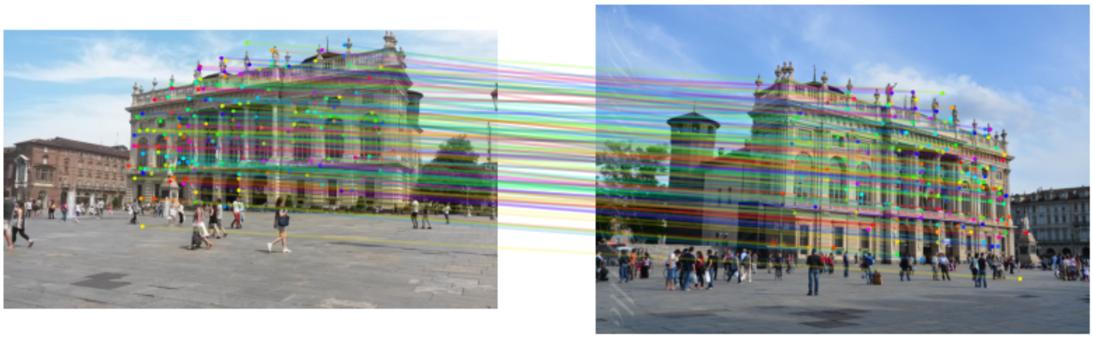


Figure 4.11: Example of keypoints extracted by SuperPoint and matched by SuperGlue. The pictures are from Turin scene train set. in the figure only 150 of the 1077 matches are showed, to have a more readable image.

5. Images, local features and matches are imported into the COLMAP database and the incremental mapper is called. There are some parameters of the mapper that can be tuned to find a tradeoff between performance and runtime. They include:

- *ba_global_images_ratio*: the growth rate in the number of images after which to perform global bundle adjustment.
- *ba_global_points_ratio*: the growth rate in the number of points after which to perform global bundle adjustment.

- *ba_global_max_num_iterations*: the maximum number of global bundle adjustment iterations.

  Some different values of these parameters have been tested, as will be shown in subsection 4.4.2. A bunch of other parameters can be fine tuned to improve the accuracy of the reconstruction, but I did not carry out many experiments because the focus of this thesis is on the retrieval step. By default during the bundle adjustment the focal length and the skew parameter are refined, while the principal point position is not. Since, as explained in 4.3, I will not use the skew for the point reprojection, I did not perform a refinement of this value, but I did it for the principal point instead. If not every image in the set shares covisibility areas with the others, connected components in the image matching graph can be created. In this case COLMAP reconstruct many sub-models, theoretically depicting different scenes, for a maximum of 50, by default. For the evaluation I used only the largest model, that usually covered almost the totality of the images. Hence, I set the *multiple_models* parameter to 0, so that only the biggest connected component is reconstructed, and the required time is reduced.

6. Once the reconstruction is done, images that have not been registered are deleted from the global descriptor database and from COLMAP database, as well as keypoints and matching related to them.

7. The 100 test images are processed as previously the training ones: global and local descriptors are extracted with the same methods used in the previous phases 2 and 3. Then pairs between test images and already registered ones are selected. Although the used method is the same, the parameters are set to $N = 20$, $k = 0$ and $min\_score = 0$ for every experiment. This is done to pick a sufficient number of pairs, so that as many images as possible are registered. The matching over these pairs is done with the same method used previously, in step 4.

8. Test images, their keypoints, and their matches are imported into COLMAP database, and the function *image_registrator* is called.

9. The output binary files of the newly updated reconstruction are converted to `txt` format, to be more easily consulted successively.

10. Points seen by test images are projected on the photos as discussed in 4.3 and the error is computed.

4.4.1 The importance of image retrieval

Let me show in this first test, the importance of using image retrieval in a SfM pipeline. It is obviously useful to reduce reconstruction time, as images are not paired exhaustively, but just a smaller number of significant pairs is taken into account. What I would like to emphasize is that it also improves the accuracy of the model. This tests are made on the Statue of Liberty scene, because the small number of images in it allows to carry out experiments without the use of retrieval in a reasonable time. The scene contains 400 train images and 100 test images. Both the retrieval results have been made with the setting $N = 40$, $k = 0$ and $min\_score = 0$, thus each image is paired with the 40 images more similar to it.

Pairing method		Reconstruction		Test set	
Retrieval	Pairs	Images	Time	Images	MRE
No retrieval	79800	207	2:06	77/100	4.886
NetVLAD	16000	231	1:02	76/100	**3.734**
Cosplace	16000	220	0:57	74/100	3.739

Table 4.1: Results on the Statue of Liberty scene, with $N = 40$, $k = 0$ and $min\_score = 0$ for the two tests with retrieval. The time is expressed in hours and minutes, and refers only to the time required for the main reconstruction. The time for registering test images and for the evaluation is not included in this value. MRE is the Mean Reprojection Error over the test images.

The time is more than halved using a retrieval method. The error is significantly lower with both the tests with retrieval, which achieve almost the same result. The number of test images registered is comparable in all the three experiments, so the MRE can be considered reliable for a comparison. There is instead a difference in the number of train images registered, which is lower in the test with exhaustive matching. This fact is quite counter-intuitive, since one can think that matching each picture with all the others, they may have higher probability to be registered. But the numerical results give another point towards using image retrieval. The lower error is probably caused by the fact that with an exhaustive pairing, SuperGlue is asked to match keypoints over images which do not share any covisibility area. Sometimes it could find similar points and mistakenly match them. The wrong matches may trick COLMAP, especially in small scenes like this one, where there is not much point redundancy. Obviously all the times showed may vary if the pipeline is run on a different hardware. They are informative only if considered with respect to times of other experiments.

4.4.2 Mapper options

Among the many parameters of COLMAP's *incremental_mapper*, the ones I set with a value different from the default one are:

- $multiple\_models = 0$
- $ba\_refine\_principal\_point = 1$
- $ba\_refine\_extra\_params = 0$

This setting holds for all the following tests on all the scenes. Then I made some experimentation on different values of global bundle adjustment parameters, in particular $ba\_global\_images\_ratio$, $ba\_global\_points\_ratio$ and $ba\_global\_max\_num\_iterations$ to see the best fitting configuration for this work. The result are presented in Table 4.2.

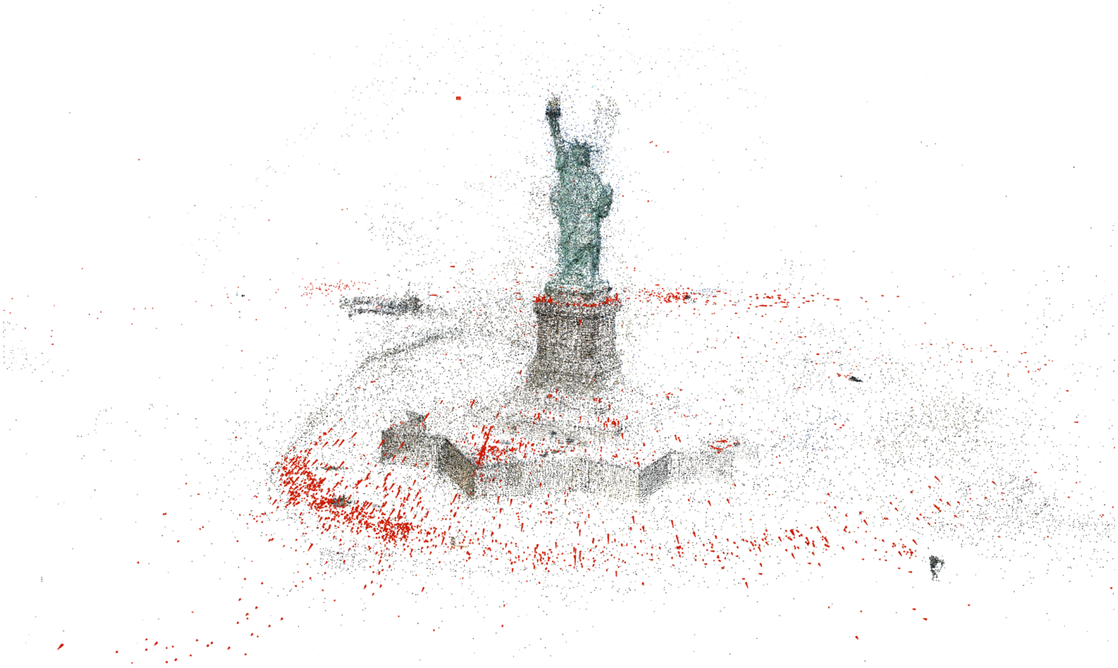


Figure 4.12: Reconstruction of the Statue of Liberty scene. This one is obtained from the model with NetVLAD and $N = 40$. The red rectangles are the cameras' pose estimation.

Mapper global BA options			Reconstruction		Test set	
Images ratio	Points ratio	Max iters	Images	Time	Images	MRE
1.1	1.1	25	1661	52:09	41/100	5.168
1.1	1.1	15	1649	24:52	40/100	5.193
1.2	1.2	25	1621	26:38	40/100	**5.156**
1.2	1.2	15	1633	20:12	41/100	5.194
1.3	1.3	25	1651	16:14	41/100	5.207

Table 4.2: Results of different values of global bundle adjustment parameters on Venice scene, containing 4000 train images and 100 test ones. All the tests are made using Cosplace for the retrieval, with $N = 20$, $k = 0$ and $min\_score = 0$.

These tests have been carried out on Venice scene for time reasons, this being the smallest set apart from the Statue of Liberty one. The image pairs are the same, selected as the 20 most similar images, based on Cosplace descriptors. The number of registered test images is quite low, but similar in all the tests. This could be caused by the small size of the dataset, together with the high noise and variety in Flickr photos. Low values of images and point ratios, make COLMAP perform global bundle adjustment more frequently. The expected result is a higher runtime and a lower error. The reconstruction time follows these expectations. The high mean reprojection error in the test with lowest ratios is quite odd, and probably can be due to the higher number of registered train images. Reducing the maximum number of iterations allows a significant reduction of the time, at the cost of a little worsening of the accuracy. Obviously these are just some sparse experiments, but these parameters could be fine tuned to find the best configuration for each specific experimentation needs. To achieve good results in an acceptable time, I chose the happy medium with both the ratios equal to 1.2 and the maximum number of iterations to 25. This configuration will be the only one used in all the following experiments.

4.4.3 Torino - Piazza Castello

The scene contains 11652 train images and 100 test ones. It is evident from a first look at the pictures that several of them do not come from “Piazza Castello” square, or even have been taken in indoor places. This will represent quite of a challenge for the reconstruction pipeline. Let us first compare the results of NetVLAD and Cosplace with different values of N.

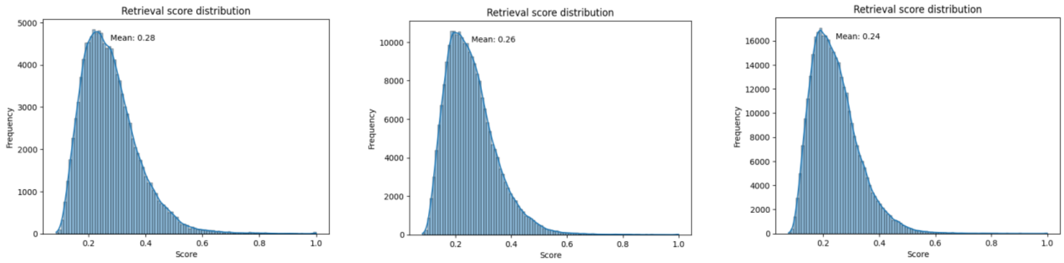
Retrieval		Reconstruction		Test set	
Extractor	N	Images	Time	Images	MRE
NetVLAD	10	2627	23:25	76/100	4.198
NetVLAD	20	5040	40:49	76/100	**3.876**
NetVLAD	30	5591	61:12	75/100	3.897
Cosplace	10	4719	32:58	87/100	4.237
Cosplace	20	5151	43:20	86/100	4.144
Cosplace	30	5187	50:34	87/100	4.224

Table 4.3: Results of different global feature extractors and different values of N on the Turin scene. In all the tests $k = 0$ and $min\_score = 0$.

From Table 4.3 it is clear the effect of N on the time required. It seems not true anyway that an higher value of pairs leads to more accurate results, probably because a lower value of N allows to a priori discard wrong matches that would only challenge COLMAP’s mapper. However if its value is too low, as in the case of $N = 10$, not only the error increases, but the number of registered images drops. This is another value to strongly take into account. Indeed most people would not say that a reconstruction with only a dozen of points perfectly triangulated is a good reconstruction, even if the error is almost zero. In these experiments NetVLAD seems to achieve better results than Cosplace in similar times.

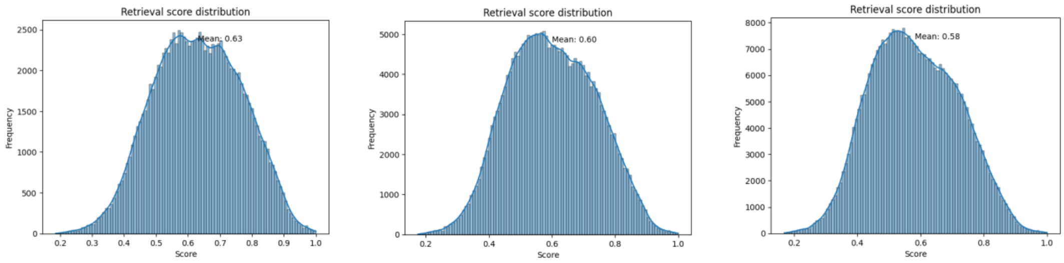
Since filtering wrong matches is crucial, as shown above, I realized that choosing the N best matches for each image is not always a good idea. If a picture is taken indoor for example, it would still be paired with N images, but all these matches would be useless and probably mistakes. Hence I thought about filtering the pairs with a low retrieval score. To do this it is useful to see the distribution of scores over the pairs selected in the experiments in Table 4.3. The relative plots are shown in Figures 4.13 and 4.14.

What can be observed is that Cosplace has higher score values, but this only depend



(a) (b) (c)

Figure 4.13: Distribution of retrieval scores based on NetVLAD over the selected pairs on Turin scene. (a) refers to the experiment with $N = 10$, where the mean score is 0.28. (b) to the one with $N = 20$ where the mean score is 0.26. (c) to the one with $N = 30$ where the mean score is 0.24. In all the three cases $k = 0$ and $min\_score = 0$.



(a) (b) (c)

Figure 4.14: Distribution of retrieval scores based on Cosplace over the selected pairs on Turin scene. (a) refers to the experiment with $N = 10$, where the mean score is 0.63. (b) to the one with $N = 20$ where the mean score is 0.60. (c) to the one with $N = 30$ where the mean score is 0.58. In all the three cases $k = 0$ and $min\_score = 0$.

on the different strategy to extract numerical descriptors and will not affect the reconstruction. The scores have almost a normal distribution between 0 and 1 and, obviously, for higher values of N the mean score is lower. To discard the pairs on the lower queue of the curve, I tried to select, from the N best matches, only the ones with a score higher than min_score. I tested some values of this parameters, as shown in Table 4.4. A more extensive testing of min_score values can be found in subsection 4.4.5 on the Venice scene.

The result on NetVLAD with $min\_score = 0.4$ must be discarded, since the low number of both train and test images makes it not comparable with other models. Clearly a too high value of this threshold leads to a loss of information and models like this are the result. With NetVLAD this strategy does not seem to help the reconstruction, while with Cosplace and $min\_score = 0.5$, accuracy and time are improved. With a 0.25 threshold on NetVLAD, the number of registered images drops and becomes similar to the test with $N = 10$. Probably 0.25 is a too high value and, indeed, the number of pairs in the two

Retrieval			Reconstruction		Test set	
Extractor	N	*min_score*	Images	Time	Images	MRE
NetVLAD	10	0.25	2627	22:02	68/100	**3.972**
NetVLAD	20	0.25	2613	34:33	72/100	3.994
NetVLAD	20	0.4	55	0:31	14/100	4.550
Cosplace	20	0.25	5235	47:14	88/100	4.283
Cosplace	20	0.5	4080	32:57	84/100	4.079

Table 4.4: Results of different *min_score* values on Turin dataset. In all tests $k = 0$.

models is similar (116520 with $N = 10$, 107185 with $N = 20$ and $min\_score = 0.25$).

I wondered if, discarding images which are not connected to the registered ones, the results would just improve or if some different images could help in some way. To discover this, I choose the model with NetVLAD, $N = 20$ and $min\_score = 0.25$, and I ran a reconstruction with only the 2613 registered images as input. The test set instead is still the same. The results are shown in Table 4.5.

Retrieval			Reconstruction		Test set	
Extractor	N	*min_score*	Images	Time	Images	MRE
NetVLAD	20	0.25	2535	27:03	71/100	4.023

Table 4.5: Reconstruction run over the only 2613 images registered in the experiment with NetVLAD, $N = 20$ and $min\_score = 0.25$. The score is here thresholded again because otherwise it would select pairs that were not in the previous model, with very low scores. This way, 83.6% of the pairs selected here, had already been selected in the old model.

The result is not as good as I expected. It achieves a significant time reduction, but the error is even higher. This seems to suggest that some wrong matches, and some images to be discarded, help COLMAP with the reconstruction. Hence I looked for a filtering to be performed on the pairs and not on the images.

Figure 4.15 shows the average number of matches for each of the 20 pairs selected for each image, in the model with NetVLAD, $N = 20$ and $min\_score = 0$. Here, with "matches" I refer to the keypoints matches, performed by SuperGlue, and not to the pictures pairs.

What is interesting here is the gap between the mean number of matches in registered images and in discarded ones. Filtering pairs based on the number of matches, would be similar to selecting only (or mainly) the registered images, which is an information otherwise achievable only running a previous reconstruction. I discarded from the pairs file all the images for which, the higher number of matches among the 20 pairs was lower than 500. Different values of the thresholds have been tested and 500, that preserves 97600 pairs, achieved the best results.

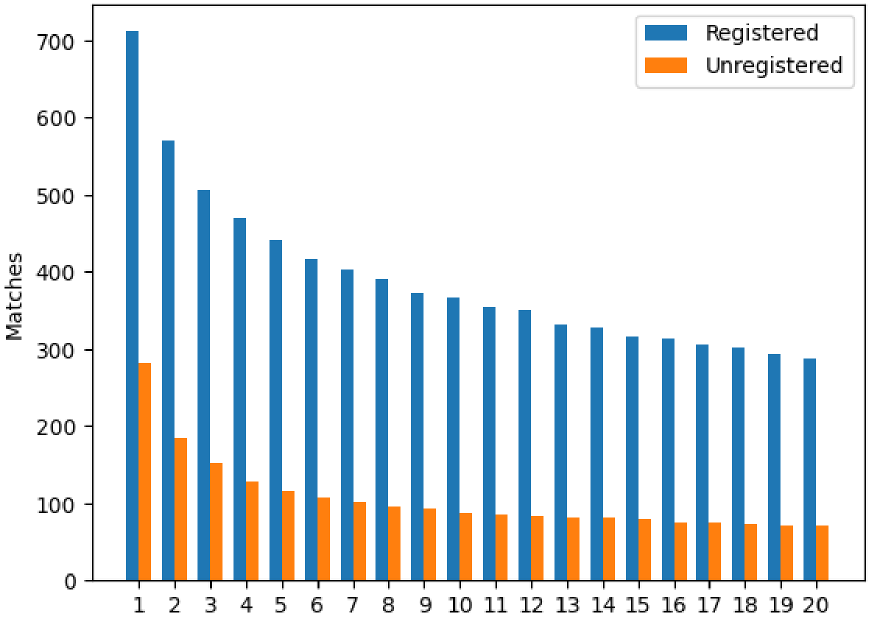


Figure 4.15: Average number of point matches for each of the $N = 20$ selected pairs. The blue columns only refer to images that have been registered in the reconstruction, while the orange ones are from discarded images. This data are relative to the experiment on Turin scene with NetVLAD global descriptors, $k = 0$, $min\_score = 0$ and SuperPoint keypoints matched with SuperGlue.

Retrieval			Reconstruction		Test set	
Extractor	N	min_score	Images	Time	Images	MRE
NetVLAD	20	0	3974	28:36	71/100	3.915

Table 4.6: Reconstruction run over filtered pairs. The images for which the best retrieval pair had less than 500 matches have been discarded.

The results are good both in terms of number of registered images and of mean reprojection error. It is essentially comparable to the simple model with NetVLAD, $N = 20$ and $min\_score = 0$, but saving approximately 12 hours.

Lastly I considered the possibility of “skipping” the pairs with the highest score. The idea behind this is that, if two images have the same view of the same place, the will have a very high retrieval score, but the will not give much information for the point triangulation. The parameter k allows to discard the k pairs with highest score for each image.

Skipping the pairs with higher scores brings an improvement when NetVLAD descriptors are used. That is not the case instead with Cosplace descriptors. The required time has a slight increase but this is understandable since, choosing pairs with lower scores

Retrieval			Reconstruction		Test set	
Extractor	N	k	Images	Time	Images	MRE
NetVLAD	15	3	4929	44:10	77/100	**3.817**
NetVLAD	20	5	4960	42:25	74/100	3.925
Cosplace	15	3	4854	31:46	84/100	4.195

Table 4.7: Results of reconstructions skipping the top k pairs for each image on Turin scene. In all the tests $min\_score = 0$.

may imply less accurate matches and more difficulties during triangulation and bundle adjustment.

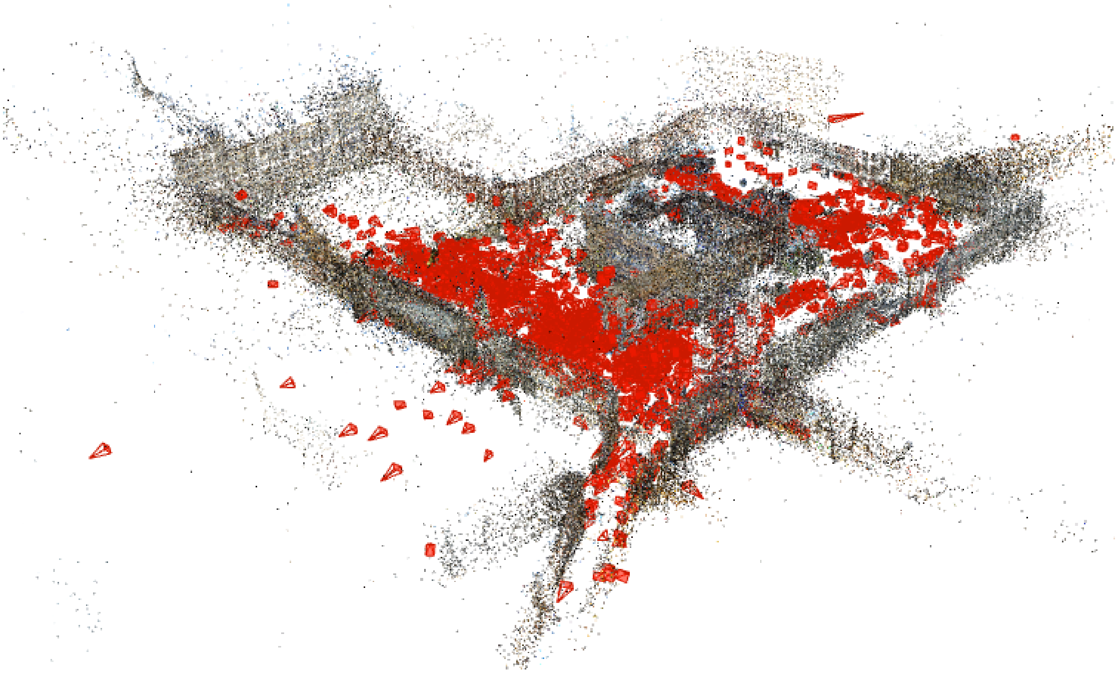


Figure 4.16: Reconstruction of the Turin Piazza Castello scene. This one is obtained from the model with NetVLAD descriptors, $N = 20$, $k = 0$ and $min\_score = 0$. The red rectangles are the cameras' pose estimation.

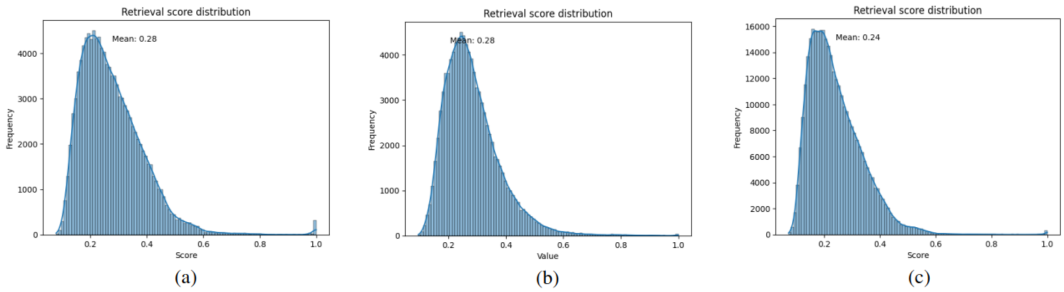
4.4.4 Barcelona - Plaça d'Espanya

This scene contains 11477 train images and 100 test ones. Here I carried out similar experiments to the ones on the previous scene.

Retrieval		Reconstruction		Test set	
Extractor	N	Images	Time	Images	MRE
NetVLAD	10	4371	39:46	78/100	4.135
NetVLAD	20	4534	44:40	79/100	4.050
NetVLAD	30	4616	67:33	78/100	4.127
Cosplace	10	4430	38:10	75/100	**3.893**
Cosplace	20	4563	51:52	75/100	3.980
Cosplace	30	4513	68:59	76/100	4.277

Table 4.8: Results of different global feature extractors and different values of N on the Barcelona scene. In all the tests $k = 0$ and $min\_score = 0$.

In this case Cosplace seems to work slightly better than NetVLAD. The number of registered images here grows with N, as expected, and the time does the same. An interesting trend is that the mean reprojection error, passing from $N = 10$ to $N = 20$, decreases with NetVLAD but increases with Cosplace. Someone could speculate that the first pairs (the ones with an higher score) are more useful with Cosplace but not with NetVLAD. This idea will be also suggested by other results in the following. Another possible explanation is the fact, observable in Figure 4.18, that some of the Cosplace pairs have a value close to 1. These pairs, which are obviously always selected, are made of almost identical photos, which give poor contribution to the reconstruction process, as already stated. But this issue appears in NetVLAD scores as well, so this is probably not the cause.



(a) (b) (c)

Figure 4.17: Distribution of retrieval scores based on NetVLAD over the selected pairs on Barcelona scene. (a) refers to the experiment with $N = 10$, where the mean score is 0.28. (b) to the one with $N = 20$ where the mean score is 0.28. (c) to the one with $N = 30$ where the mean score is 0.24. In all the three cases $k = 0$ and $min\_score = 0$.

Table 4.9 shows the results of using thresholds on the retrieval score. The results of similar experiments without thresholds are also reported as a reference.

In both cases the threshold allows to reduce the reconstruction time. On the other hand, the registered images are also reduced, but Cosplace can register approximately 500 images more than NetVLAD. However its MRE is higher, while with NetVLAD is

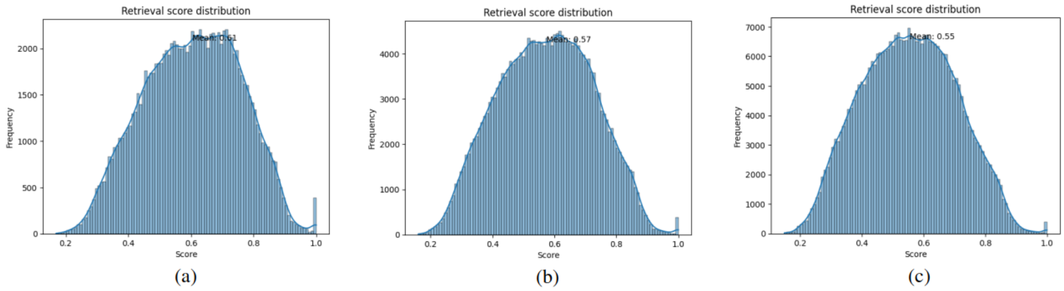


Figure 4.18: Distribution of retrieval scores based on Cosplace over the selected pairs on Barcelona scene. (a) refers to the experiment with $N = 10$, where the mean score is 0.61. (b) to the one with $N = 20$ where the mean score is 0.57. (c) to the one with $N = 30$ where the mean score is 0.55. In all the three cases $k = 0$ and $min\_score = 0$.

Retrieval			Reconstruction		Test set	
Extractor	N	min_score	Images	Time	Images	MRE
NetVLAD	20	0	4534	44:40	79/100	4.050
NetVLAD	20	0.25	3370	41:17	72/100	4.028
Cosplace	20	0	4563	51:52	75/100	**3.893**
Cosplace	20	0.6	3877	48:58	72/100	3.977

Table 4.9: Results of different min_score values on Barcelona dataset. In all tests $k = 0$.

essentially the same. Someone could argue that the thresholds are different and so the tests are not comparable. Actually these values have been chosen to discard almost the same fraction of pairs. Indeed, 54.6% of NetVLAD pairs have a score lower than 0.25, while 51.7% of Cosplace ones are under 0.6.

Finally Table 4.10 shows the results obtained varying the value of k.

Here NetVLAD, skipping the 5 best matches achieves a significant decrease in the reprojection error, which can be considered an improvement even if the number of registered images is slightly lower. On the other hand Cosplace, that reached the best results in the basic test, is essentially not able to perform the reconstruction if the first 5 pairs are skipped. Indeed its model has only 8 images and could not register any test image. This suggests, again, that the top matches selected with Cosplace are very important and informative, while the one based on NetVLAD are not. Discarding them and considering pairs with a lower score can downright increase performance.

4.4.5 Venezia - Piazza S. Marco

This scene contains 4000 train images and 100 test ones. Being the smallest scene, the time needed for an average experiment is significantly lower. For this reason, some more

Retrieval			Reconstruction		Test set	
Extractor	N	k	Images	Time	Images	MRE
NetVLAD	10	0	4371	39:46	78/100	4.135
NetVLAD	10	5	4294	40:30	75/100	3.929
NetVLAD	15	3	4424	41:09	76/100	3.967
Cosplace	10	0	4430	38:10	75/100	**3.893**
Cosplace	10	5	8	8:37	0/100	0

Table 4.10: Results of reconstructions skipping the top k pairs for each image on Barcelona scene. In all the tests $min\_score = 0$.

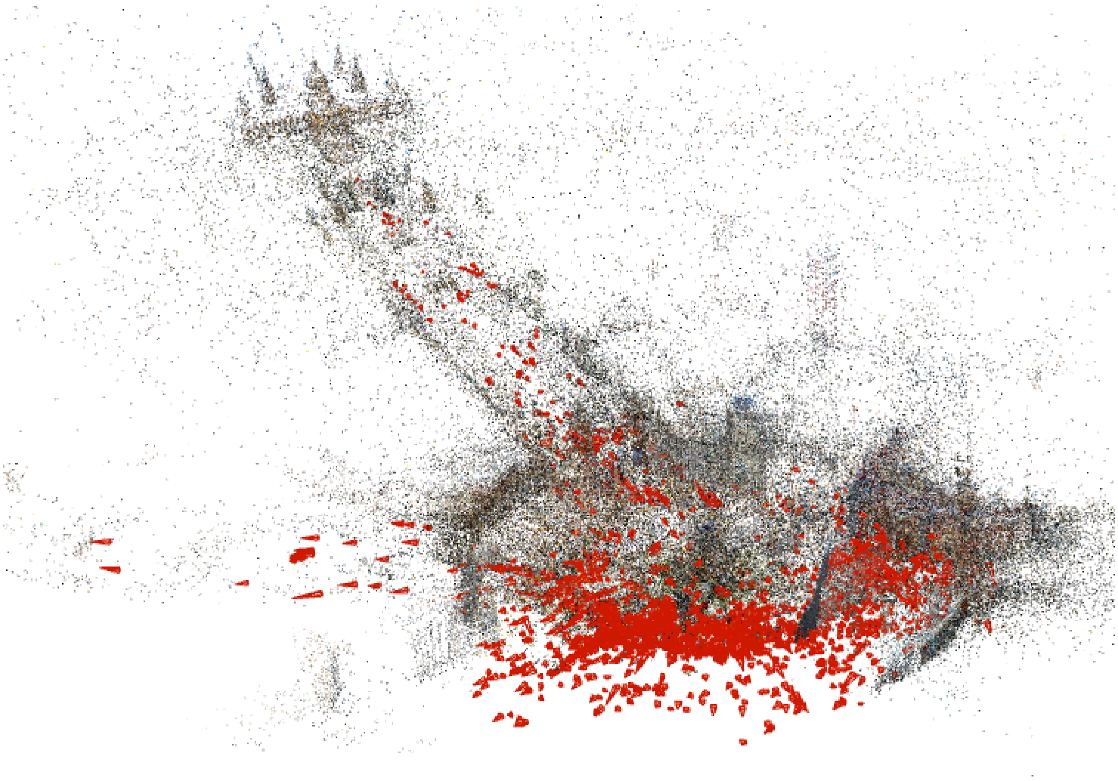


Figure 4.19: Reconstruction of the Barcelona Plaça d'Espanya scene. This one is obtained from the model with NetVLAD descriptors, $N = 20$, $k = 0$ and $min\_score = 0$. The red rectangles are the cameras' pose estimation.

tests have been conducted on this dataset, especially regarding the retrieval score threshold. The noise in the images is approximately the same as in the previous two scene, and indeed the registered images are on average around the 40% of the input ones. Since this scene is smaller however, the number of registered test images is quite low. The mean reprojection error can still be used as an informative metric, because all the tests have a similar number of test images, but probably it would be a good idea to enlarge the test set or to change the retrieval strategy for the test images registration, to avoid giving too much weight to a single photo.

As in the previous scenes, let us begin with a comparison of some basic strategies, in Table 4.11.

Retrieval		Reconstruction		Test set	
Extractor	N	Images	Time	Images	MRE
NetVLAD	10	1573	23:26	38/100	5.392
NetVLAD	20	1652	30:47	40/100	5.291
NetVLAD	30	1685	34:22	41/100	5.197
Cosplace	10	1588	25:52	40/100	5.315
Cosplace	20	1621	26:38	38/100	**5.156**
Cosplace	30	1641	33:32	40/100	5.485

Table 4.11: Results of different global feature extractors and different values of N on the Venice scene. In all the tests $k = 0$ and $min\_score = 0$.

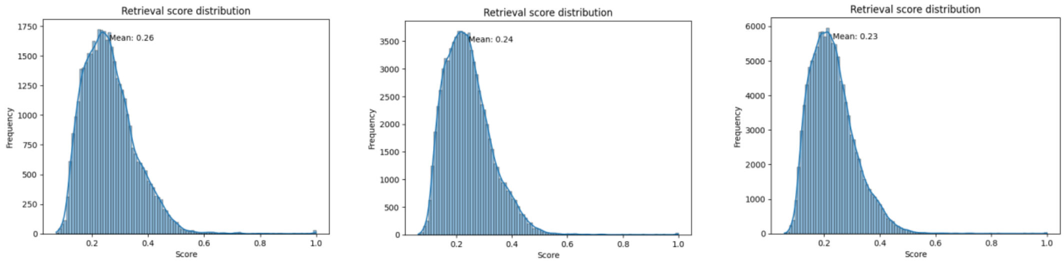
The trend of reconstruction time and of registered images against the variations of N is aligned to the expectations. NetVLAD and Cosplace achieve similar results, with no one outperforming the other in general. What is interesting to observe is that NetVLAD reaches its best with $N = 30$, while Cosplace with $N = 20$. This fact seems to suggest another time that NetVLAD needs some pairs with lower score, while for Cosplace the top scored are the most important ones. The fact that the errors are higher than in the other scenes should not concern, because, as I told in the previous sections, the MRE is consistent if used to compare tests with the same test images and the same keypoints. Otherwise, it becomes not much informative as COLMAP's mean reprojection error.

Figures 4.20 and 4.21 show the distribution of the retrieval scores of the selected pairs for NetVLAD and Cosplace descriptors, with different values of N.

A more extensive experimentation on the minimum score threshold have been carried out on this scene. The results will be presented in two separated tables, Tables 4.12 and 4.13, for the seek of clarity.

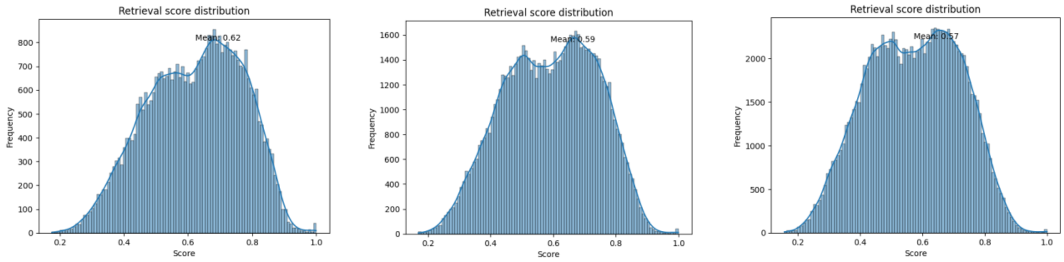
With NetVLAD descriptors, the thresholds 0.25 and 0.30 should be discarded since they register few images and have a very high MRE. As min_score increases, the error decreases, has its minimum in $min\_score = 0.15$, and then rises again. The minimum value can be considered the same as the model with $min\_score = 0$, even if here it is slightly lower, but it should still be preferred since it requires only half the time.

With Cosplace descriptors there is less degeneration of the error with high thresholds. The time required decreases more slowly than with NetVLAD. The trend of the errors is the same as with the other descriptors: it increases, then reaches its minimum in $min\_score = 0.45$, and rises again. These results are consistent, as the fraction of discarded pairs is similar in both cases. With NetVLAD, the best threshold (among the tested ones) is $min\_score = 0.15$, which is higher than 14.6% of the pairs' scores. With



(a) (b) (c)

Figure 4.20: Distribution of retrieval scores based on NetVLAD over the selected pairs on Venice scene. (a) refers to the experiment with $N = 10$, where the mean score is 0.26. (b) to the one with $N = 20$ where the mean score is 0.24. (c) to the one with $N = 30$ where the mean score is 0.23. In all the three cases $k = 0$ and $min\_score = 0$.



(a) (b) (c)

Figure 4.21: Distribution of retrieval scores based on Cosplace over the selected pairs on Venice scene. (a) refers to the experiment with $N = 10$, where the mean score is 0.62. (b) to the one with $N = 20$ where the mean score is 0.59. (c) to the one with $N = 30$ where the mean score is 0.57. In all the three cases $k = 0$ and $min\_score = 0$.

Cosplace, the value $min\_score = 0.45$ is higher than 17.5% of the pairs' scores. But not all the possible values have been tested and, in this range, even a little variation has a quite big influence on the number of selected pairs. For example, values of 0.16 for NetVLAD and of 0.435 for Cosplace would discard exactly the same fraction of pairs. One can infer that the minimum MRE is obtained discarding approximately 16% of pairs.

Lastly Table 4.14 shows some experiments with $k = 5$, that is skipping the 5 pairs with the higher retrieval score.

From these results it is clear how, also in this case, the top pairs are more important for Cosplace but not for NetVLAD. With the former extractor, skipping 5 pairs rises the error and leads to less registered images. With the latter, a higher reconstruction time is rewarded with a significant reduction of the MRE. It is also interesting to compare this result with the experiment using NetVLAD, $N = 20$ and $k = 0$, shown in Table 4.11. Here the pairs from the 1-st to the 20-th are selected, while with $N = 15$ and $k = 5$ the

Retrieval		Reconstruction		Test set	
N	*min_score*	Images	Time	Images	MRE
20	0	1652	30:47	40/100	5.291
20	0.10	1657	28:43	39/100	5.323
20	0.15	1642	14:10	41/100	**5.268**
20	0.20	1553	13:38	40/100	5.298
20	0.25	146	3:34	20/100	9.084
20	0.30	124	2:29	15/100	12.018

Table 4.12: Results of different *min_score* values on Venice dataset with NetVLAD descriptors. In all tests $k = 0$.

Retrieval		Reconstruction		Test set	
N	*min_score*	Images	Time	Images	MRE
20	0	1621	26:38	40/100	5.156
20	0.30	1673	22:20	40/100	5.306
20	0.40	1627	23:56	41/100	5.344
20	0.45	1619	22:51	39/100	**5.153**
20	0.50	1566	21:53	40/100	5.203
20	0.60	1344	18:44	38/100	5.857
20	0.70	995	13:11	38/100	5.751

Table 4.13: Results of different *min_score* values on Venice dataset with Cosplace descriptors. In all tests $k = 0$.

ones from the 6-th to the 20-th. Surprisingly, matching fewer pairs, even if the discarded ones are the best ones according to NetVLAD, allows to register approximately the same number of images in the same time, but resulting in an error decrease from 5.291 to 5.150.

Just to make a brief discussion about the time, as already told the reported values refer only to the reconstruction process, that is to the steps from 2 to 5 of Section 4.4. The evaluation process on average requires approximately 15 minutes. This is almost the same for all the experiments because the test set contains always 100 images, and the retrieval parameters are always set to $N = 20$, $k = 0$ and *min_score* $= 0$, so the number of pairs is always 2000. Of the reconstruction time, about the 15-20% is needed for feature extraction and matching. The remaining major fraction is required by the COLMAP's mapper. As shown in Figure 4.23, the growth of the time is very fast when there are few pairs. Then it slows down somewhat for a number of pairs between 60000 and 130000, and afterwards it rises more than linearly.

4.4.6 Robustness to keypoints changes

In this last subsection I would like to present some experiments where the local descriptors have been extracted with SIFT and matched with a Nearest Neighbour algorithm. The

Retrieval			Reconstruction		Test set	
Extractor	N	k	Images	Time	Images	MRE
NetVLAD	15	0	1630	25:16	40/100	5.350
NetVLAD	15	5	1631	31:01	40/100	**5.150**
Cosplace	15	0	1601	26:23	40/100	5.279
Cosplace	15	5	1584	17:45	38/100	5.396

Table 4.14: Results of reconstructions skipping the top k pairs for each image on Venice scene. In all the tests $min\_score = 0$. The tests with $N = 15$ and $k = 0$ have been evaluated for a more consistent comparison.

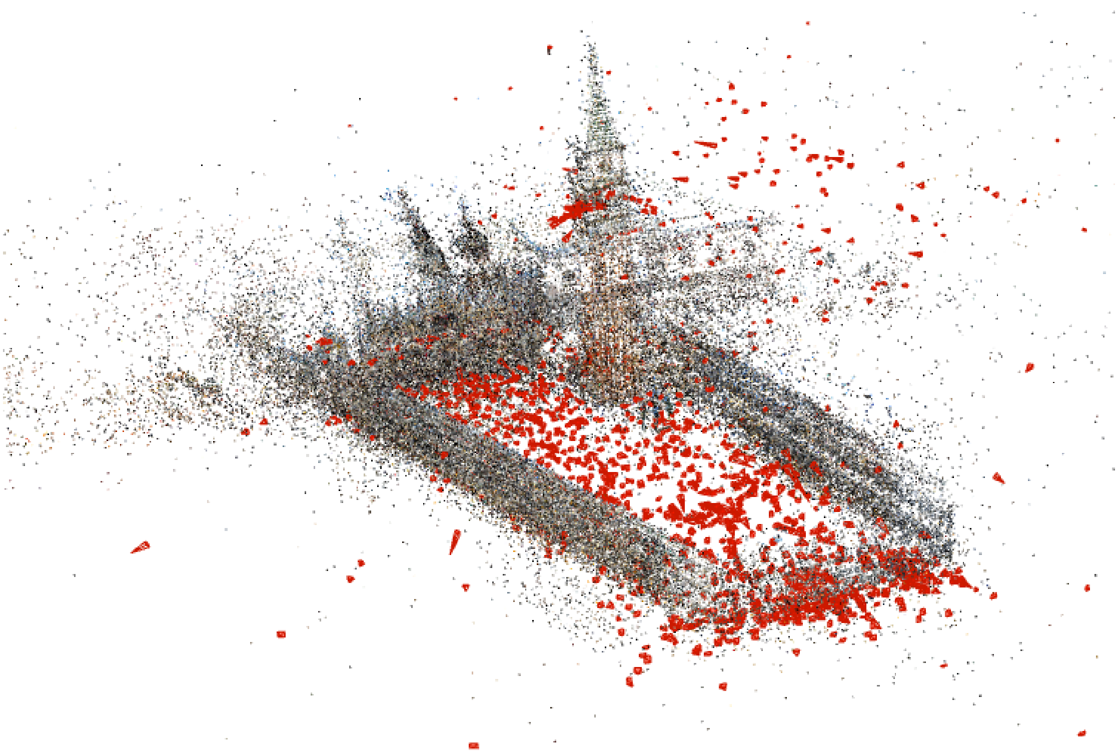


Figure 4.22: Reconstruction of the Venice Piazza San Marco scene. This one is obtained from the model with NetVLAD descriptors, $N = 20$, $k = 0$ and $min\_score = 0$. The red rectangles are the cameras' pose estimation.

idea is not to compare models with different keypoints, as this would be unfeasible with this evaluation strategy. Indeed, as I told previously, the mean reprojection error is a consistent metric as long as the experiments to compare share the same keypoints and the same test images. The aim of this test is to see if the most interesting observations made on the previous results, with SuperPoint descriptors, are also valid with other keypoints. The experiments shown in Table 4.15 are chosen to highlight some of the facts observed

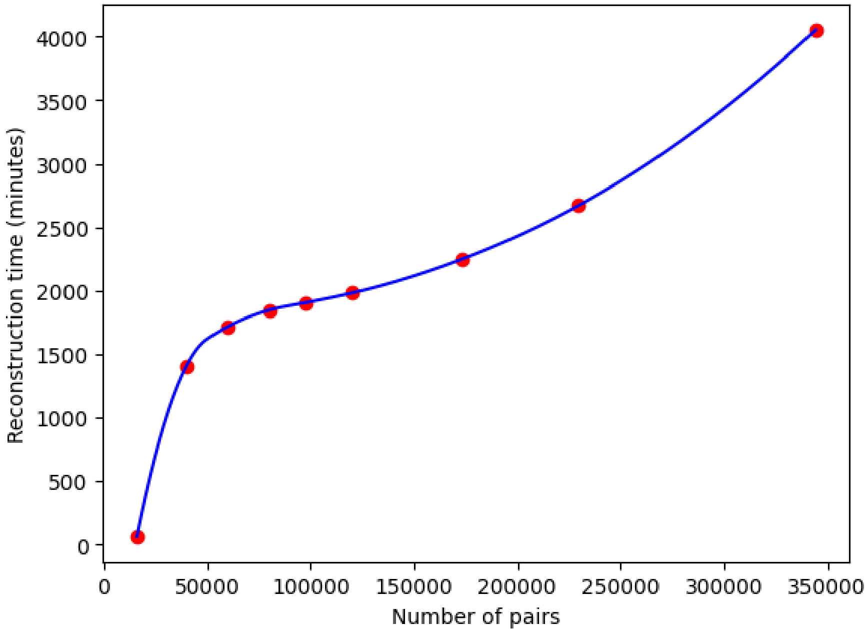


Figure 4.23: Plot of the required reconstruction times against the number of retrieval pairs considered. The red dots are experimental observations from all the scenes and different values of N. The blue line is just an interpolation line for readability. Experiments where $k \neq 0$ or $min\_score \neq 0$ have not been selected for this plot because only discarding the best or the worst pairs has a different effect on the time than just reducing the number of pairs. The worst pairs usually require more time for the triangulation and more iterations of the bundle adjustment.

in the previous subsections. All of them are conducted on the Venice scene.

Retrieval				Reconstruction		Test set	
Extractor	N	k	min_score	Images	Time	Images	MRE
NetVLAD	10	0	0	4	0:59	0/100	0
NetVLAD	15	0	5	900	28:30	25/100	4.744
NetVLAD	20	0	0	966	21:43	25/100	5.466
NetVLAD	20	0	0.15	994	14:59	26/100	4.950
Cosplace	10	0	0	844	19:37	23/100	**4.737**
Cosplace	20	0	0	1013	21:59	25/100	4.633

Table 4.15: Results of various retrieval strategies on the Venice scene. The local descriptors are extracted by SIFT and matched with a Nearest Neighbour algorithm.

In the above results there are 3 interesting behaviours to be observed. The first one is that the threshold $min\_score = 0.15$ on the test with NetVLAD and $N = 20$ reduces both the runtime and the error. A similar observation was made with SuperPoint local descriptors, even if there the decrease in the MRE was mild, while here is remarkable. In addition Cosplace improves the accuracy result as N increases from 10 to 20. The same behaviour was observed with the other keypoints. Lastly, skipping the 5 top scored pairs and taking the following 15 is better, in terms of MRE, than taking the top 20 matches, when using NetVLAD descriptors. This fact was true on SuperPoint descriptors as well, and confirms the intuition that the first pairs are not much informative with NetVLAD. The experiment with NetVLAD and $N = 10$ failed, probably because the dataset is quite small, SIFT extracts less keypoints than SuperPoint, and only the first 10 pairs are not sufficient, since, again, the top ones are not very useful with NetVLAD.

Hence, in general, the observation made on SuperPoint descriptors are true also on SIFT ones. Thus these behaviours can be considered robust to keypoints chances.

Chapter 5

Conclusions

This chapter will contain the final remarks regarding this thesis, including a discussion on the results, limitations and insights for future works.

The main goal was to explore the contribution of Visual Place Recognition on a Structure from Motion pipeline. The evaluation strategy used to this aim turned out to be consistent through experiments. Between the two tested global descriptors extractors, there is not one which significantly outperformed the other, but with no doubt the use of some retrieval strategy reduces both reconstruction time and mean reprojection error. The number of retrieval matches to be taken into consideration should be tuned on the number of images in the dataset, but, working on strongly noisy scenes as the ones used in this work, the top 20 matches are usually sufficient. A lower number would run the risk of not being able to perform a reconstruction with a decent number of registered images. A higher one would bring too much noise among the matches, affecting the accuracy of the model and slowing down the process.

The threshold on the retrieval score can be useful to reduce the required time. A value which can discard approximately the 15-16% of the pairs with the lower scores, is able to achieve results that are comparable with the ones obtained without using a threshold, but even halving the reconstruction time. A fine tuning of this parameter may allow to speed up the reconstruction process without loosing accuracy.

The skipping of the best scored pairs may allow a reduction in the reprojection error. This can be observed mainly when using NetVLAD descriptors, while with Cosplace ones this is usually not true. This highlights a difference in the pair ranking of these two extractors. NetVLAD requires to "go deeper" in the scores list. The matches with a high score result to be less informative for the reconstruction. This fact furthermore is confirmed by other experiments, such as the ones on the number N of pairs to be considered. With Cosplace descriptors, this observation does not hold true. In this case the pairs with higher score are the most important ones, and skipping them leads to an increase in the error. This difference could be explained by the different approach used by the two methods. NetVLAD is based on a positive mining strategy, which gives importance to the picture pose. On the other hand, Cosplace is designed to reward images taken in the same place, even if their poses, i.e. orientation and position, are different. The result is that the top matches based on NetVLAD are share almost all the image area of covisibility, while this is not necessarily true with Cosplace. A Structure from Motion pipeline clearly needs covisibility across pictures, but it also needs different points of view, to perform an effective triangulation of points. Hence, too similar images may not cause mistakes, but neither do they help the reconstruction.

The combination of a (not too aggressive) score threshold and of a top pairs skipping strategy on NetVLAD descriptors, may increase the accuracy of the reconstruction, allowing to keep limited the runtime.

The observations made seem to be robust to changes in the keypoints and in their matching, at least when these tasks are entrusted to SIFT and a Nearest Neighbour algorithm. It would be interesting to run some experiments on indoor scenes too, but datasets of this type are usually very small or strongly structured, and thus do not fully exploit the benefits from retrieval.

Many experiments have been conducted, but many more would have been interesting to carry out. The time constraints force me to end here the experimentation, but with a more extensive testing, maybe on different scene, probably other interesting aspects could be discovered. It would be also interesting to test other global descriptor extractors for the retrieval step. NetVLAD and Cosplace are well known and well performing algorithms, but some experimentation on more recent state-of-the-art extractors, such as SALAD (Izquierdo and Civera [2024]), would be surely interesting to do. Not only it may outperform the two older methods, but from the results some other aspects of the behaviour of SfM algorithms may be inferred.

The paved way is the one that leads to a custom-designed retrieval method, which exploits the main benefits of each existing algorithm to outperform them in this specific field. This is possible because Visual Place Recognition is applied to a wide variety of contexts and tasks. For this reason, a general purpose algorithm must aim to achieve decent results in almost every application, making adaptability its strength. However an algorithm that works optimally for visual geo-localization, may have poor performance is such a specific task as SfM. A method suited for this task could bring significant improvements in the scene reconstruction. For a work of this type though, a dataset with known ground truth would be necessary. If the reconstruction evaluation can be done even without known camera poses, following the strategy used in this thesis, the knowledge of the exact position and orientation of each photo would be crucial to train a custom retrieval method. The lack of city-scale datasets with these information, but still with heterogeneous images in terms of viewpoints, is maybe the main limitation towards this goal.

Bibliography

Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79, 2009. doi: 10.1109/ICCV.2009.5459148.

R. Arandjelovic and A. Zisserman. All about vlad. *Proceedings IEEE/CVF Conference on Computer Vision Pattern Recognition (CVPR)*, page 1578–1585, 2013.

R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *Proceedings IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2018.

Relja Arandjelović, 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 (CVPR)*, 2016.

A. Babenko and V. Lempitsky. Aggregating deep convolutional features for image retrieval. *Proceedings IEEE/CVF International Conference on Computer Vision (ICCV)*, page 1269–1277, 2015.

A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. *Proceedings ECCV*, page 584–599, 2014.

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

Fabio Bellavia, Jiri Matas, Dmytro Mishkin, Luca Morelli, Fabio Remondino, Weiwei Sun, Amy Tabb, Eduard Trulls, Kwang Moo Yi, Sohier Dane, and Ashley Chow. Image matching challenge 2024 - hexathlon, 2024. URL https://kaggle.com/competitions/image-matching-challenge-2024.

Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking Visual Geo-localization for Large-Scale Applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

R. Chellappa, G. Qian, and S. Srinivasan. Structure from motion: sparse versus dense correspondence methods. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 2, pages 492–499 vol.2, 1999. doi: 10.1109/ICIP.1999.822945.

Z. Chen, O. Lam, A. Jacobson, and M. Milford. Convolutional neural network-based place recognition. In *Australasian Conference on Robotics and Automation (ACRA)*, 2014.

Ashley Chow, Eduard Trulls, HCL-Jevster, Kwang Moo Yi, lcmrll, old ufo, Sohier Dane, tanjigou, WastedCode, and Weiwei Sun. Image matching challenge 2023, 2023. URL https://kaggle.com/competitions/image-matching-challenge-2023.

M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research (IJRR)*, 27(6): 647–665, 2008.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proceedings IEEE Computer Society Conference on Computer Vision Pattern Recognition*, page 886–893, 2005.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018. URL https://arxiv.org/abs/1712.07629.

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: Part 1. *IEEE Robotics & Automation Magazine (RA-M)*, 13(2):99–110, 2006.

D. Galvez-Lòpez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics (TRO)*, 28(5):1188–1197, 2012.

Sourav Garg, Tobias Fischer, and Michael Milford. Where is your place, visual place recognition? In *In proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Stuart I. Granshaw. Structure from motion: origins and originality. *The Photogrammetric Record*, 33(161):6–10, 2018. doi: https://doi.org/10.1111/phor.12237. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/phor.12237.

S. Haner and A. Heyden. Covariance propagation and next best view planning for 3d reconstruction. *ECCV*, 2012.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

Qi Hu, Jianxin Luo, Guyu Hu, Weiwei Duan, and Hui Zhou. 3d point cloud generation using incremental structure-from-motion. *Journal of Physics: Conference Series*, 1087 (6):062031, sep 2018. doi: 10.1088/1742-6596/1087/6/062031. URL https://dx.doi.org/10.1088/1742-6596/1087/6/062031.

Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition, 2024. URL https://arxiv.org/abs/2311.15937.

H. Jegou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. *Proceedings IEEE Conference on Computer Vision Pattern Recognition*, page 3310–3317, 2014.

H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. *Proceedings IEEE Computer Society Conference on Computer Vision Pattern Recognition*, page 3304–3311, 2010.

Fua P. Lepetit V., Moreno-Noguer F. Pnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections nature. page 133–135, 1981.

D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1999.

Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 9, 2021.

E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giro i Nieto. Bags of local convolutional features for scalable instance search. *Proceedings of the 30th ACM International Conference on Multimedia*, page 327–331, 2016.

A. Oliva and A. Torralba. *Building the gist of a scene: The role of global image features in recognition*, pages 23–36. Elsevier, Amsterdam, The Netherlands, 2006.

Onur Ozyesil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion, 2017. URL https://arxiv.org/abs/1701.08493.

M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. *Proceedings IEEE International Conference on Computer Vision (ICCV)*, page 91–99, 2015.

F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. *Proceedings IEEE Computer Society Conference on Computer Vision Pattern Recognition*, page 3384–3391, 2010.

Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions, 2018. URL https://arxiv.org/abs/1707.09092.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

S. Schubert, P. Neubert, and P. Protzel. Fast and memory efficient graph optimization via icm for visual place recognition. *Robotics: Science and Systems (RSS)*, 2021.

Stefan Schubert, Peer Neubert, Sourav Garg, Michael Milford, and Tobias Fischer. Visual place recognition: A tutorial. *IEEE Robotics & amp; Automation Magazine*, page 2–16, 2024. ISSN 1558-223X. doi: 10.1109/mra.2023.3310859. URL http://dx.doi.org/10.1109/MRA.2023.3310859.

Sivic and Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings IEEE/CVF International Conference on Computer Vision (ICCV)*, 2003.

N. Snavely, S. M. Seitz, , and R. Szeliski. Photo tourism: exploring photo collections in 3d. *SIGGRAPH*, 2006.

G. Tolias, R. Sicre, and H. Jãgou. Particular object retrieval with integral max-pooling of cnn activations. *Proceedings of 4th International Conference on Learning Representations (ICLR)*, page 1–5, 2016.

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, pages 298–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008. ISSN 1572-2740. doi: 10.1561/0600000017.

Michał J. Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient, 2020. URL https://arxiv.org/abs/2006.13566.

S. Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London*, page 405–426, 1979a.

S. Ullman. *The Interpretation of Visual Motion*. MIT Press, 1979b.

C. Valgren and A. J. Lilienthal. Sift, surf & seasons: Appearancebased long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149–156, 2010.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Fine-tuning cnn image retrieval with no human annotation. *IEEE Conference on Computer Vision and Pattern Recognition*, page 5265–5274, 2018.