TUM School of Computation, Information and Technology
TECHNISCHE UNIVERSITÄT MÜNCHEN

DEEP LEARNING MEETS VISUAL LOCALIZATION

Qunjie Zhou

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:		Prof. Dr. Matthias Nießner
Prüfer*innen der Dissertation:	1.	Prof. Dr. Laura Leal-Taixé
	2.	Prof. Dr. Marc Pollefeys
	3.	Prof. Dr. Jiri Matas

Die Dissertation wurde am 22.12.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 21.06.2023 angenommen.

Preface

The first time I touched on the topic of visual localization was in 2017 when I started my master's thesis under the supervision of Laura Leal-taixé and Torsten Sattler, who later became my PhD supervisor and mentor. After exploring the world of visual localization for 5 years, I still enjoy it very much. One charm of this topic is that it lies in the intersection of several other important computer vision topics, such as image retrieval, image matching, SLAM and SfM. Another characteristic I enjoyed greatly in this topic is the freedom to provide various solutions depending on the use cases. Much interesting research has emerged during the past decades in this field, focusing on different aspects of a localization systems, such as accuracy, computation efficiency, generalization, long-term maintenance and privacy perseverance. Therefore, I am delighted to give a brief introduction to visual localization through this thesis and meanwhile share you with our recent research findings and contributions to this topic. Yet, before the formal start, I would like to give thanks to all of the important people that have enriched my journey of pursuing a PhD.

First of all, my biggest thanks to Laura, my dear supervisor, for everything you have taught me since we met. You have witnessed my growth in the past 5 years. Thank you for being more than a supervisor but also a good friend of mine. I enjoyed our discussions about research, work, life and the future. Every time I could not move forward, you always manage to bring me courage, confidence and cheers, which are the essential fuel for my PhD journey. I also want to give special thanks to Torsten, my dear mentor, it was a great time doing research together with you. More importantly, without your guidance from the beginning when I knew nothing about visual localization, I could have totally missed this very interesting topic. I want to give my sincere thanks to all of my dear colleagues: Aljosa, Andreas, Aysim, Franzi, Guillem, Illi, Jenny, Mark, Maxim, Orcun, Patrick, Sergio, Tim. I feel very lucky to meet you all at DVL, and my PhD years are furnished with those memorable time that I have spent with you during the conferences, submission deadlines and retreats, Christmas dinners, journal club sessions and all kinds of outdoor activities. In addition, I want to give my special thanks to Sabine, our dear secretary, for your efforts in helping us to handle all administrative things all of the time. Another special thanks and respect to Quirin, our dear administrator. Your incredible and amazing work in maintaining vast computing resources and solving numerous technical problems has safeguarded every doctoral candidate within our chair to survive submission deadlines.

Finally, I want to thank my family, Tong, Wei, Jinjing and my friends in the bible study group for your unconditional love and companionship. Thank you all for cheering and enjoying with me during the highs, and encouraging, comforting and supporting me during the lows.

Qunjie Zhou
Munich, Dec 2022

ABSTRACT

Localizing an agent in 3D environment is a fundamental ability to enable spatial-aware intelligent systems such as robot navigation and tracking, autonomous driving and interactive Mixed Reality applications. We choose the path of visual localization where we determine our locations in the form of camera poses based on visually perceived images. In the past decades, the extremely rapid development of deep learning has brought innovations to visual localization which has led to significant improvement of the existing methods meanwhile opened up several new options for tackling localization. However, one big challenge coming with deep neural networks is how to interpret the behavior of a learned model when they are not reaching our expectations. Therefore, this thesis focuses on investigating and understanding the potentials and limitations of the existing learning-based localization as well as developing new data-driven solutions to address the current challenges encountered by the state-of-the-art localization methods.

Purely data-driven relative pose regression has recently been proposed as an alternative to the classical feature matching-based solution, yet it shows a large performance gap from the non-learning-based methods. As our first contribution, we propose an essential matrix-based localization framework to analyze the reason behind their limited generalization and accuracy. Our experiments diagnose that the issue is coming from the pose regression layers instead of the image feature extractor, which contributes important insights into future work towards better relative pose-based localization.

Recently emerging correspondence networks learn end-to-end image matching inside a single network but are suffering from low matching resolution due to the memory bottleneck. As our second contribution, we present a new perspective to estimate correspondences in a *detect-to-refine* manner, which aims to elegantly improve the matching resolution to the pixel level. Our learned refinement network based on direct matches regression significantly improves the performance of correspondence networks on image matching and localization, and also generalizes across multiple matching methods and datasets.

The well-established structure-based localization relies on visual descriptors to establish matches between a query image and a 3D point cloud. While being highly accurate, it encounters multiple challenges in storage demands, privacy concerns and long-term maintenance. As our last contribution, we go beyond the classical visual descriptor matching and match keypoints solely relying on its geometric information. Our experimental evaluation confirms its potential and feasibility for real-world localization, which opens the door to future efforts towards more general and scalable structure-based localization.

In addition to presenting our contributions, we also serve this thesis as a short review on several topics of interest, covering image retrieval, image matching and visual localization. In the exploding mass of literature, we briefly introduce the important works of these topics since proposed, which hopefully makes their development trajectories more traceable.

CONTENTS

Part I

Introduction and Background

1 Introduction

1.1 Visual Localization

1.1.1 Motivation

Localization refers to the task of determining *where an agent is* in a target map based on the query data obtained from sensors. *Global Positioning System* (GPS) is one of the most widely used localization systems, where the query data are GPS signals sent by an agent (device) and the map is the whole earth described in the geographic coordinate system, *i.e.*, latitudes and longitudes. In fact, in our daily life, we already heavily rely on GPS-enabled applications such as Google Maps[1], to conveniently localize ourselves, share locations with friends and navigate to a destination. Despite GPS being fairly mature over decades of development, there are several well-known limitations. Since it requires reliable signal transmission for localization, it is not applicable to or functioning well in areas where GPS signals can not be properly sent and received, such as mountain areas and underground stations. In the urban area, the localization accuracy of GPS is typically in meters, *e.g.*, 5-10 meters [LaM+05], for outdoor regions and might not work at all for indoor environments. Those issues have motivated the birth of other localization techniques that rely on different types of sensors. For example, *Wi-Fi Positioning System* (WPS) [LZP11; SLM14] is usually used for the indoor environment when GPS techniques are not sufficient meanwhile Wi-Fi signals are available.

Different from GPS and WPS, *visual localization* [SLL02; RC04; Irs+09; SLK16; Sar+19] uses digital cameras as sensors which are easily accessible from personal mobile devices such as smartphones, tablet and laptops. Given a query image, a visual localization system estimates the position and orientation of the camera capturing the query image relative to a 3D scene map. In comparison to GPS, visual localization aims for higher localization accuracy, *e.g.*, within a meter. However, it is not competitive but rather complementary to GPS techniques. It can be combined with GPS to handle large-scale scenes, where GPS determines world-widely a coarse location and then visual localization refines the location with higher precision. Such high-precision localization technique is especially required by many Mixed Reality applications [Art+09; CKM08; Mid+14; Ven+14; Lyn+15; Art+11], where there are needs to overlay augmented contents accurately to the real scene contents. In addition, visual localization is a vital component in various vision-based intelligent systems such as autonomous driving [Hen+19] and robot navigation [Don+09; Irs+09; WIB11; Lim+12], while being closely related to other computer vision tasks such as Structure-from-Montion [Wu13; Wu+11; SF16] and Simultaneous Localization And Mapping [SLL01; MMT15; Dav+07].

[1] https://www.google.com/maps

1.1.2 A Brief History

One of the earliest works on this topic is proposed by Se *et al.* [SLL02], where they directly match SIFT [Low99] features against a 3D model reconstructed with SLAM and later estimate a geometry transformation inside a RANSAC scheme [FB81]. Another early visual localization system proposed by Robertson & Cipolla [RC04] performs two-view matching to identify a nearby view w.r.t. a query view and then estimates the relative transformation between the query and its nearby reference view to finally recover the query pose. Its follow-up work [ZK06] generalizes this prototype by eliminating the requirement that each image needs to be dominant with a building facade. These early works became the pioneers of *structure-based* [SLL02] (section 4.6) and *relative pose-based* [RC04; ZK06] (section 4.5) localization respectively. Concurrently, with the development in image retrieval (section 2.3) techniques [SZ03; Csu+04; NS06], several works [SH+04; SBS07] propose *image retrieval-based* localization (section 4.5) where they cast the localization problem as an image retrieval problem. Given a query and a database of geo-tagged images, the location of the query is approximated by the location of its top-similar retrieved database image. This formulation attracts a vast body of follow-up work [JDS08; JDS09; Avr+10; ZS10; KSP10] due to its simplicity, efficiency and scalability. Inspired by [GL06; SL04] that use Structure-from-Motion (SfM) model for object pose estimation, Arnodl *et al.* [Irs+09] maintains a sparse SfM point cloud as a 3D scene representation where each 3D point is associated with its descriptor that was used for reconstruction. In fact, the emergence of stable large-scale SfM techniques [Fra+10; SSS06; Aga+11] has also promoted the development of visual localization. It allows one to conveniently build an image database with camera pose labels as well as reconstruct a compact and informative scene representation in the form of a 3D point cloud, opening up the possibility of large-scale localization. Nowadays, modern SfM softwares such as colmap [SF16; Sch+16] have become the standard tools to build scene representation and ground truth for visual localization.

When it comes to the rival of machine learning, mostly deep learning, tons of computer vision tasks such as image classification [Den+09; Den+09], image retrieval [Bab+14; Ara+16; Azi+15; RTC18], semantic segmentation [LSD15] and object recognition [Ren+15], were reformulated into their learning-based versions, leading to the advanced state-of-the-art in each topic. Similarly, visual localization also experienced a revolution with emerging learning techniques. The first absolute pose regression (section 4.3) method, *i.e.*, PoseNet, was proposed by Kendall *et al.*, to directly regress camera pose from a single image, and later was improved by numerous follow-up works [Wal+17; KC17; Bra+18; Xue+20; SFK21]. Another novel formulation of scene coordinate regression (section 4.4) was proposed by Shotton *et al.* [Sho+13] to implicitly match 2D pixels to their corresponding 3D scene points via regression, followed by a standard pose estimation from the 2D-3D matches using PnP solvers [KSS11; Gao+03]. A large number of latter works have been proposed to improve the accuracy [Guz+14; Bra+17], efficiency [Bra+16; BR21] and adapt the formulation to work for large scenes [TZ00] or multiple scenes [Yan+19; Li+20a]. Building on top of the modern learned image retrieval [Ara+16; RTC18] and learned image matching [DMR18; Sar+20; Sun+21], the latest structure-based localization sets the state-of-the-art localization performance in the challenging long-term localization benchmarks [Sat+18]. Compared to the most reliable and accurate structure-based localization, absolute pose regression and scene coordinate regression techniques have the benefit of being relatively more efficient in runtime and memory. We leave more thorough review of the existing visual localization methods in chapter 4 and its related topic image matching in chapter 8.

1.2 Contributions and Outline

This cumulative thesis comprises three full-length first-author publications covering two closely related topics image matching and visual localization. These publications are the result of joint work with Torsten Sattler, Sérgio Agostinho, Aljoša Ošep, Prof. Marc Pollefeys and Prof. Laura Leal-Taixé. In table 1, we show a complete summary of all works published during the whole doctoral program, which includes three other second-author publications covering the topics of visual localization, cross-view localization and text-based localization. In all of our research works, one core interest for us is to explore how to properly leverage deep learning techniques in those topics. We now brief the motivation and conclusion of our works.

Research projects involved in this thesis. In EssNet [Zho+20], we developed an essential matrix-based framework that supports fair comparison between various relative pose estimation methods for visual localization. Within this framework, we compare different ways of leveraging deep learning to estimate relative pose from an image pair. From our experiments, we found that the popular purely data-driven method, relative pose regression, struggles to generalize to unseen scenes, while using explicit matching for relative pose estimation leads to better generalization and accuracy. Based on this finding, we turn our attention from direct pose regression to explicit image matching via deep learning and conducted our next project.

In Patch2Pix [ZSL21], we take inspiration from modern object detection methods and propose to perform hierarchical image matching in a *detect-to-refine* paradigm, where we first detect coarse matches at patch-level and then refine match accuracy within a matched patch pair. We adopt the newly emerging correspondence network NCNet [Roc+18] to densely search a match proposal within two images and develop a novel patch2pix refinement network to directly regress pixel-level matches from multi-scale deep features. The refinement network effectively improves multiple types of coarse match proposals, showing the idea of *detect-to-refine* is a promising direction for achieving highly accurate matches.

Despite the huge success of structure-based localization via visual descriptor matching, we observe the challenges in storage, privacy, maintenance for real-world large-scale localization, which all stem from the reliance on visual descriptors. In GoMatch [Zho+22], we seek for geometric-based matching to bypass the need for visual descriptors, as an orthogonal direction to address those challenges in visual localization. We build upon the existing BPNPNet [Liu+20; CLG20] to boost its robustness and accuracy for realistic visual localization, from almost not working on real-world data to similar competitive performance compared to the state-of-the-art APR methods. Our work points out an encouraging future direction to advance structure-based localization for real-world large-scale scenes.

Other research projects. In UnderstandAPR [Sat+19], we provide mathematical theory for absolute pose regression based on which we analyze its limitations in performance and use cases, we also draw conclusions for future research directions for APR techniques. Despite not being included in this thesis, this project comprehends our understanding of the popular pose regression techniques. We give more detailed introduction and discussion about absolute pose regression in section 4.3.

In [Tok+21], we investigate the task of geo-localization in a cross-view setting, *i.e.*, localizing a ground-view query image against an aerial-view satellite map. While closely related to classical visual localization, the main different technical challenge in this scenario is how to handle the extreme cross-view difference in appearance and viewpoints. In Text2Pos [Kol+22], we take one bold step further and propose a new localization task, text-based localization, where the user query data is a textual description of the surroundings of the place of interest. We experimentally prove the feasibility of this new task with a general coarse-to-fine text-based localization baseline. These two projects [Tok+21; Kol+22] have broadened the author's horizons and expanded the author's knowledge of different research potentials and challenges in the general topic of localization. We refer the readers to full versions of the papers for more details.

Thesis outline. In the following contents of Part I of this thesis, we continue our background introduction by first providing some fundamental knowledge in chapter 2. Next, we dive into the core topics in this thesis which are image matching in chapter 8 and visual localization in chapter 4. We browse the historical development of these two topics through a literature summary. After that, we go into Part II where we summarize our main publications, *i.e.*, EssNet in chapter 5, Patch2Pix in chapter 6 and GoMatch in chapter 7. Finally, we summarize our works, propose open challenges that we identified and sketch out interesting directions for future research.

[Sat+19] Torsten Sattler, **Qunjie Zhou**, Marc Pollefeys, and Laura Leal-Taixé.
Understanding the Limitations of CNN-based Absolute Camera Pose Regression.
In *CVPR 2019*.

[Zho+20] **Qunjie Zhou**, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixé.
To Learn or Not to Learn: Visual Localization from Essential Matrices.
In *ICRA 2020*.

[ZSL21] **Qunjie Zhou**, Torsten Sattler, and Laura Leal-Taixé.
Patch2pix: Epipolar-guided Pixel-level Correspondences. In *CVPR 2021*.

[Tok+21] Aysim Toker, **Qunjie Zhou**, Maxim Maximov, and Laura Leal-Taixé.
Coming Down to Earth: Satellite-to-street View Synthesis for Geo-localization.
In *CVPR 2021*.

[Kol+22] Manuel Kolmet, **Qunjie Zhou**, Aljoša Ošep, and Laura Leal-Taixé.
Text2Pos: Text-to-Point-Cloud Cross-Modal Localization.
In *CVPR 2022*.

[Zho+22] **Qunjie Zhou***, Sérgio Agostinho*, Aljoša Ošep, and Laura Leal-Taixé.
Is Geometry Enough for Matching in Visual Localization? (*equal contribution)
In *ECCV 2022*.

Table 1: Publication Summary. For completeness, we state all of our works published during the doctoral program while leaving those are not included as part of this thesis in gray.

2 Foundations

In this chapter, we introduce fundamental background knowledge that helps to fully understand the later contents in this thesis. In the first section 2.1, we present the concept of camera model that defines the projective geometry between points in a 3D world and pixels in a 2D image, followed by epipolar geometry in section 2.2 that defines the geometrical relationship between two images observing common scene contents, which serves as an important tool to solve relative camera motion between two images by establishing 2D correspondences. Finally, in section 2.3 we introduce the topic of image retrieval and briefly summarize its evolution since proposed. The image retrieval is highly related to the next two chapters. While chapter 8 describes how to extract local features for image matching, image retrieval shows how to use local features to globally represent an image for content searching. On the other hand, image retrieval techniques have been widely applied to visual localization in various manners, playing a significant role in large-scale localization, which we will describe in detail in chapter 4.

2.1 Pinhole Camera Model

A camera model is an important mathematical tool to help understand and analyze the working principles and properties of a real-world camera, which is one of the most fundamental sensors in computer vision. In this thesis, we stick to the *pinhole camera* model which is the simplest camera model and has been widely used in many computer vision tasks including visual localization.

Camera coordinate system. The above fig. 1 is an illustration of a pinhole camera model. For simplicity, we consider the (virtual) image plane which is positioned in front of the camera center instead of the actual film plane located behind the camera center with flipped x- and y-axis. Let's now define the *camera coordinate system* in 3D, where its origin is the camera center C and the line from the camera center perpendicular to the image plane is the principal axis or depth axis. The intersection of the image plane and the principal axis is the principal center $(o_x, o_y)^T$, which is also the center of the image plane. In pinhole setting, the distance between the camera center and the image plane is the focal length f.

Camera central projection. As shown in the fig. 1, a 3D point $P = (X, Y, Z)^T$ defined w.r.t. the camera coordinate system captured by the pinhole camera is mapped into a 2D point $p = (x, y, f)^T$ on the image plane. The two points are related by two similar triangles, from which we can derive $(x, y)^T = (fX/Z, fY/Z)^T$. Thus we define

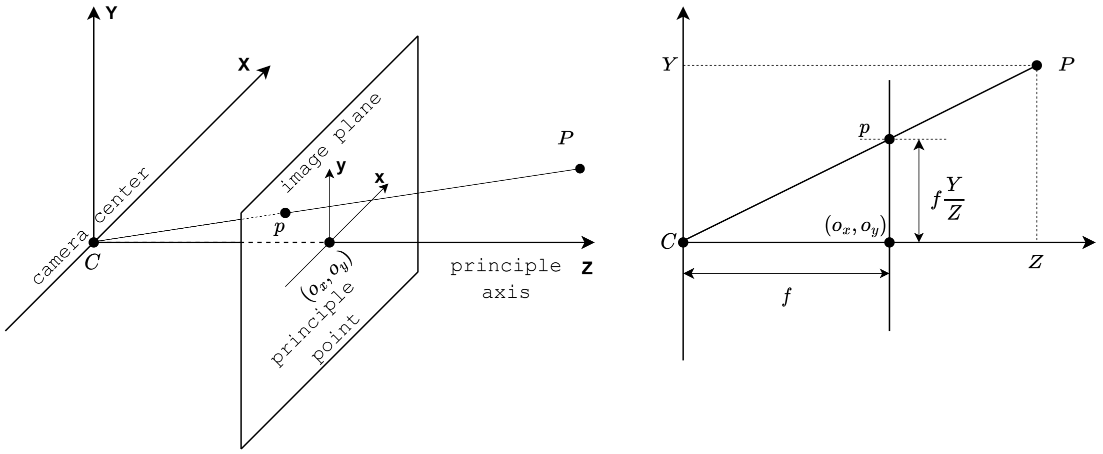


Figure 1: Pinhole camera model illustration.

the central projection of the pinhole camera as a mapping:

$$\pi \colon \mathbb{R}^3 \to \mathbb{R}^2 \ ; \ P \mapsto p = \pi(P) = \begin{pmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{pmatrix} . \tag{1}$$

The eq. (1) can be written in the form of matrix multiplication with homogeneous coordinates:

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} . \tag{2}$$

Camera intrinsic matrix. In the above, we assume the principal point to be the origin of the image plane. However, the origin of digital image coordinates is typically at the lower-left corner of the image. In addition, digitalizing continuous image plane into discrete pixels might leads to pixel units with different scales compared to the physical unit. Taking those effects into account, we write eq. (3) in a more general form:

$$p_h = \begin{pmatrix} fs_xX + Zo_x \\ fs_yY + Zo_y \\ Z \end{pmatrix} = \begin{bmatrix} fs_x & 0 & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K \begin{bmatrix} I & 0 \end{bmatrix} P_h \tag{3}$$

where s_x, s_y are the scaling factors along x- and y-axis and $*_h$ denotes a homogeneous form of a vector. And K describes the internal setting of a camera model, thus usually called the *camera intrinsic matrix.*

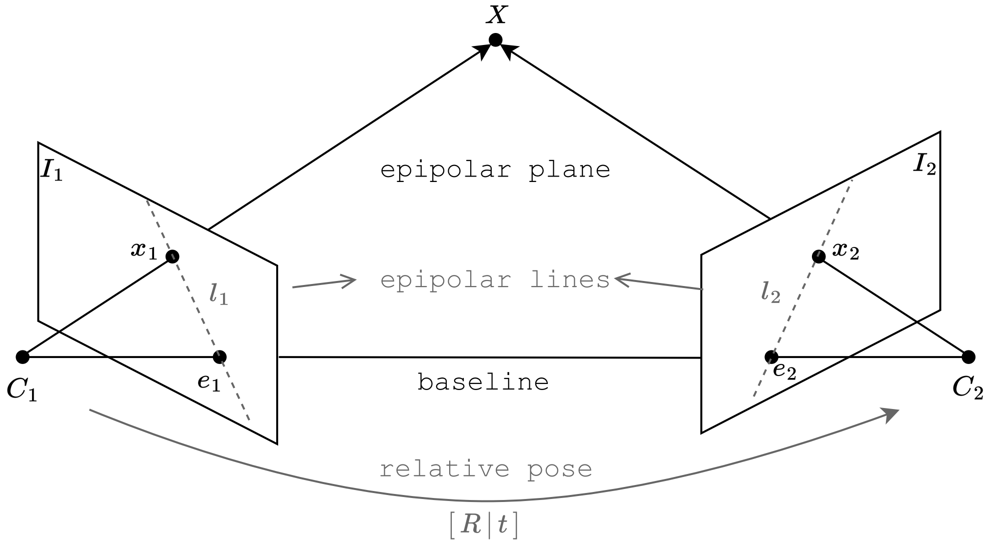


Figure 2: Epipolar geometry illustration.

Camera extrinsic matrix. As the camera moves, the camera coordinate system changes accordingly w.r.t. to a global world coordinate system. Therefore, to fully model the mapping from a 3D point defined globally to a 2D image pixel, we still need to further introduce the *camera extrinsic matrix* which is composed by a translation vector $t \in \mathbb{R}^3$ and a 3×3 rotation matrix $R \in SO(3)$. It transforms a 3D point P_w from the world coordinates to the camera coordinate system (P) with the following equation:

$$P = RP_w + t = \begin{bmatrix} R & t \end{bmatrix} \begin{pmatrix} P_w \\ 1 \end{pmatrix}. \tag{4}$$

Combining both the intrinsic and extrinsic matrices, we show a world 3D point P_w is mapped to a 2D image pixel p by:

$$p = K \begin{bmatrix} R & t \end{bmatrix} \begin{pmatrix} P_w \\ 1 \end{pmatrix} = H \begin{pmatrix} P_w \\ 1 \end{pmatrix}. \tag{5}$$

In the end, we obtain a 3×4 camera projection matrix H that includes both the intrinsic and extrinsic parameters of a pinhole camera model.

Camera pose. With the extrinsic parameters R and t, we can compute the position of a camera center in the world coordinates by $C_w = -R^{-1}t$. As R and t can describe the camera orientation and position globally, they are also called *(absolute) camera pose.*

2.2 Epipolar Geometry

Epipolar geometry [HZ03; Sze22; FM90] describes the geometric relations between two images and the scene structure, *i.e.*, 3D points, commonly visible in both images, assuming the two views are captured with a pinhole camera. It establishes the most common solution for relative camera motion estimate via image pair correspondences [HZ03; Nis04; LH06; Kuk+17], and thus is also a fundamental step in SfM. Geometric distances defined based on an essential matrix and a fundamental matrix are also used as objectives for learning image matching [Wan+20b; Li+22]. For this thesis, we leveraged epipolar geometry in both EssNet [Zho+20] (chapter 5) and Patch2Pix [ZSL21] (chapter 6). We now describe in short the basic concepts in epipolar geometry.

Terminologies. The fig. 2 shows a minimal setup to illustrate the epipolar geometry defined w.r.t. a 3D point X and the two images I_1 and I_2 with camera centers at C_1 and C_2. The lines (C_1, X) and (C_2, X) connecting the two optical centers and the 3D points intersect the two image planes at $x1$ and $x2$, the two corresponding projections of X in the two images. The line connecting the camera centers C_1 and C_2 is called *baseline*, which crosses the two image planes at two *epipoles* e_1 and e_2. The baseline and the 3D point define the *epipolar plane*, which intersects the image planes in the two *epipolar lines* l_1 and l_2.

Epipolar constraint. Given a known pixel x_1 with unknown depth, its corresponding 3D point has to lie on the ray goes from C_1 to x_1. Therefore, with epipolar geometry, we know the correspondent pixel x_2 of x_1 in the second image, has to lie on the epipolar line l_2. This allows one to examine whether a predicted correspondence is consistent with epipolar geometry.

Now we present the mathematical derivation of epipolar geometry. For simplicity, we assume the camera intrinsic matrix K for both images is known as an identity matrix. By choosing the first frame as the reference frame, its camera pose is given by a rotation $R_1 = I$ and a translation $t_1 = 0$. Given the relative camera motion defined by a relative rotation R_{12} and a translation t_{12} going from I_1 to I_2, the two projections x_1, x_2 can the expressed as:

$$\lambda_1 x_1 = X \ , \lambda_2 x_2 = RX + t \implies \lambda_2 x_2 = \lambda_1 R x_1 + t \tag{6}$$

where λ_1 and λ_2 are the depth of x_1 and x_2. We then get rid of t by applying a cross product of t as $t \times t = 0$. As a cross product can be represented in matrix multiplication, $a \times b = [a]_\times b$, where $[a]_\times$ is the skew-symmetric matrix of the vector a, we multiply $[t]_\times$ to the whole equation:

$$\lambda_2 [t]_\times x_2 = \lambda_1 [t]_\times R x_1 + [t]_\times t = \lambda_1 [t]_\times R x_1. \tag{7}$$

As $[t]_\times x_2$ represents the normal direction of the plane defined by t and x_2, we project it onto x_2 and derive *epipolar constraint* [HZ03]:

$$x_2^T \lambda_2 [t]_\times x_2 = 0 = x_2^T \lambda_1 [t]_\times R x_1 \implies x_2^T [t]_\times R x_1 = x_2^T E x_1 = 0 \tag{8}$$

where $E = [t]_\times R$ is the *essential matrix* that encodes the relative transformation between the two cameras. We further extend eq. (8) for unknown camera intrinsic matrices denoted as K_1 and K_2:

$$x_2^T K_2^{-T} [t]_\times R K_1^{-1} x_1 = x_2^T F x_1 = 0 \tag{9}$$

We call $F = K_2^{-T}[t]_\times RK_1^{-1}$ the *fundamental matrix* that encodes both the camera intrinsic and extrinsic parameters.

Applications. The fundamental matrix maps a point x_1 in the one image I_1 to its corresponding epipolar line l_2 in the other image I_2 by $l_2 = Fx_1$. Similarly, we obtain the epipolar line l_1 of a point x_2 as in $l_1 = F^T x_2$. According to the epipolar constraint, if x_1 and x_2 are perfectly matched, they should exactly lie on l_1 and l_2. Therefore, the distance between x_1 and l_1 and the distance between x_2 and l_2 can be used to measure the quality of the correspondence (x_1, x_2).

On the other hand, given a set of correspondences matched from a pair of images, one can estimate the essential matrix or the fundamental matrix using N-point solvers [Nis04; LH06; Ste+08; HL12] from which one recovers the relative camera motion between the image pair [HZ03]. Since the epipolar constraints are defined up to scale, the extracted relative translation only tells the direction of the actual translation vector.

2.3 Image Retrieval

The general task of image retrieval (IR) is to search and identify relevant images from a database according to query data in the form of a textual description or an image, which describes the contents of interest for searching. Image retrieval is a long-standing task that was naturally motivated by the need to efficiently find information of interest from image collections [Hal+06]. The early solutions to image retrieval are based on textual annotations where images are described with keywords or captions and retrieved based on the text descriptions [CF80; RFS88]. However, due to the high cost of annotating images with text descriptions, later researchers switch to exploiting visual cues for image retrieval which is called content-based image retrieval (CBIR) [Hal+06].

A CBIR system extracts and stores visual features for all database images in advance and then the same type of features are extracted from a query to compute similarity against every database image based on the feature distances [SZ03; Ara+16; RTC18]. The retrieval candidates are those database images with top-ranked similarity to the query. After the initial retrieval, then post-processing techniques such as spatial verification [Phi+07; She+13; SAC19; Tei+19; RTC18; CAS20; Noh+17; TYO21; Lee+22] and query expansion [Chu+07; Qin+11; TJ14; RTC18] can boost the accuracy.

To effectively represent an image is the core of a CBIR system. The common image representations for retrieval either represent an image as a set of local descriptors called Bag-of-Words (BoW) [SZ03; Csu+04] or as a compact global descriptor that can be aggregated from handcrafted [JZ14; GKS13; Jég+11; Jég+10; PD07] or deep features [Azi+15; BL15; KMO16; RTC18; TSJ15; Gor+16; Gor+17; Ara+16; Tor+15; CAS20] or directly extracted from deep neural networks [Bab+14].

Bag-of-Words. The influential work from Sivic and Zisserman [SZ03] firstly presents a complete system based on BoW representation for image retrieval, which latter has become a common practice of the following BoW-based image retrieval methods [CPM09; Chu+07; Phi+07; Mik+13; Cao+10; Zho+10]. A typical BoW representation for an image is constructed in three steps: i) A set of keypoints are firstly identified from the image. ii) Local descriptors are then extracted at the keypoint locations. iii) Finally the extracted descriptors are quantized by replacing each descriptor with its most similar visual word from a codebook [SZ03]. A codebook is typically learned offline from a large image database, where descriptors extracted from all images are quantized with K-means clustering [SZ03] into K visual words that compose the codebook. The step i) and ii) are normally done with handcrafted feature detection algorithms [Low04; AZ12; PCM09] such as SIFT [Low04]. We will present more thorough introduction about local features in chapter 8. To perform image retrieval with BoW representations, the main question is how to compute the similarity between two images in BoW. Following the idea from the document retrieval technique, a BoW image (analogy to a document) is represented by a frequency vector computed based on the visual word occurrences and the database is structured as an inverted file system to allow efficient comparison between BoWs [SZ03; Phi+07; Jég+11].

Aggregating local features. Despite the usage of inverted list structure for searching [SZ03], BoW-based image retrieval suffers from high memory requirement (32GB for 1M images) [Per+10], decreasing searching accuracy and efficiency with increasing vocabulary size [Jég+10; Jég+11; Per+10]. In order to make image retrieval practical for large-scale database with millions of images, various aggregation techniques [JZ14; GKS13; Jég+11; Jég+10; PD07] are proposed to condense a set of local features into a compact vector representation. The concept of local feature aggregation for image retrieval is also widely applied to later rising deep features which we introduce in the next paragraph.

Deep features for retrieval. The revolution of deep learning, specifically Convolutional Neural Networks (CNNs) has largely changed the state of image retrieval that was dominated by BoW-based methods. CNNs show prominently powerful capability in learning image representations that have surpassed classical hand-crafted features in many computer vision tasks such as image classification, object detection and semantic segmentation. The first works [Bab+14; Gon+14] that apply deep CNN features to image retrieval simply take the output of a fully-connected layer as a global representation of an image [Bab+14] or a region in the image [Gon+14]. The latter works improve the deep representation by aggregating the feature map after a convolution layer into a single vector, which is achieved using different pooling mechanisms [Azi+15; BL15; KMO16; RTC18; TSJ15; Gor+16; Gor+17; Noh+17; CAS20; Wan+22] or more sophisticated aggregation kernels [Ara+16; Tor+15; Hau+21; Tei+19]. There are several commonly used loss functions to train deep features for image retrieval, including contrastive loss [CHL05], triplet losse [SKP15; HA15], average precision loss [Rev+19a] and Arcface loss [Den+19]. It has also been investigated to learn features from other tasks such as classification [MC21; Bab+14; BMC22; Noh+17] during training, and apply the learned features directly to retrieval during inference.

3 Image Matching

In this chapter, we introduce image matching which is one of the core topics this thesis contributes to through our publication Patch2pix (*c.f.* chapter 6). This topic can better prepare readers for the next visual localization topic whose development is highly related to image matching. We start this chapter with an overview of the topic in section 3.1, followed by two sections of literature review on the traditional handcrafted local features in section 3.2 and the modern learning-based ones in section 3.3. In the last two sections, we talk about the recently emerging research directions on learning descriptor matching and correspondence filtering in section 3.4 and end-to-end learned matching pipeline in section 3.5.

3.1 Overview

Image matching refers to the task of establishing pixel or patch correspondences between two images. It is a fundamental step in a mass of downstream computer vision tasks such as image registration and stitching, image retrieval, SfM, visual SLAM and image-based localization. It is also a long-standing topic that researchers pay close attention to and have achieved great progress in the past few decades, especially with the huge success of deep learning. Among the massive literature on this rich topic, we focus more on learning-based approaches to highlight the impact of deep learning on image matching, meanwhile we still point out earlier influential works. We refer readers to several related reviews [MS05; TM+08; GHT11; SQ17; CH18; Ma+21] for more systematic and comprehensive summary.

Pipeline. Given a pair of images, a typical image matching pipeline consists of the following three key components: i) local feature extraction via keypoint detection and description, ii) descriptor matching and iii) matching outlier filtering. We will introduce each of the key components in detail in the following text.

Local feature. In the context of image matching, local feature extraction is usually also termed keypoint detection and description. A local feature is an image pattern that differs from its immediate neighborhood [TM+08]. It can be extracted or recognized from image properties such as intensity, color and texture. A keypoint can be defined as the geometric location where the local feature emerges. To localize a keypoint, one needs to analyze its local neighborhood of pixels, which is a local region centered at the keypoint location with a specific size and shape. Later, a vector representation of the keypoint is computed from this region based on some measurements. The descriptor capturing the local information of the keypoint is then used in downstream tasks such as local feature matching and image retrieval.

Good local features for matching. Depending on the applications, whether a type of local feature is good can be evaluated from multiple aspects including repeatability, distinctiveness, locality, quantity, accuracy and efficiency [TM+08]. It is widely recognized that repeatability, accuracy and efficiency are the three most important criteria when using the features for image matching. Repeatability indicates whether features detected on the same scene contents can be repeatably found in images observing the scene taken under different viewing conditions. Accuracy indicates whether two matched keypoints accurately correspond to the same location in the scene. Efficiency is related to the time required to compute keypoints and descriptors as well as the size of descriptors which directly influences the computation speed and memory requirement of matching.

3.2 Handcrafted Local Feature

3.2.1 Handcrafted Detectors

Early handcrafted detectors commonly identify local features by searching corner features defined by intersections of straight lines and straight corners [TM+08]. A corner can be detected via exploring various low-level image statistics such as first-order intensity derivatives [Mor77; HS+88; För94; Shi+94; Bau00], intensity comparison [SB97; TH98; Rub+11] and second-order derivatives [Low99; MS01; MS04; Low04; BTG06] where the core principle is to localize feature of high variance. In the early works, Harris corner detector [HS+88] is the most famous one that has been extended in numerous works [SM96; ZWT99; MS02; Bau00; Tri04; BTV11; BM22]. Among a vast body of handcrafted detectors, SIFT [Low04] is still the most popular and widely recognized one. Despite being relatively more expensive to compute than other detectors optimized for speed [RD06; RPD08; BTG06; Rub+11], SIFT provides highly discriminative and accurate keypoints that are commonly preferred by various matching-involved applications. While most researchers opt for deep learning to devise better keypoint detectors (*c.f.* section 3.3.1), HarrisZ+ [BM22], an upgraded HarrisZ corner detector [BTV11], is recently presented to speak up for the potential of handcrafted detectors for image matching. They show that the gap between handcrafted and deep learned detectors can be reduced when adopting some modifications and used jointly with other learned descriptors, yet still being relatively less accurate w.r.t. the state-of-the-art data-driven approaches.

3.2.2 Handcrafted Descriptors

Since keypoint detection only provides the geometrical representation of a local feature which is simply a two-dimensional vector local to a specific viewpoint, it is not sufficient to perform matching with that alone. A following description stage is required to provide a more informative representation based on a local neighborhood of the keypoint, which is expected to be compact, discriminative and invariant to image transformations for robust matching. In a way analogous to handcrafted keypoint detection, classical keypoint descriptors are commonly constructed by analyzing low-level image information through various heuristic strategies based on gradient statistics [Low99; Low04; BTG06; MY09; TW09; TLF09; DS15; AZ12], local binary pattern statistics [OPM02; HPS09; GPM10;

Che+13], local intensity comparison [Cal+10; LCS11; Rub+11; AOV12] and local intensity ordinal information [Tan+09; WFW11; Wan+15]. Among those categories, gradient-based floating descriptors such as SIFT [Low04], its variants [AZ12; BTG06] are still widely adopted nowadays in wide-baseline matching, while binary descriptors such as BRIEF [Cal+10], FREAK [AOV12] and ORB [Rub+11], being relatively efficient yet less distinctive, are used more for small-baseline matching.

3.3 Learning-based Local Feature

3.3.1 Learned Detectors

The first attempts of learning detection leverage machine learning techniques such as a decision tree [LF06; Ozu+09; RD06], shallow neural networks [CR97; DKS95], where the common idea is to learn classifiers to help to select accurate corners. FAST [RD06] is usually considered as a representative of an early machine learning-based detector. It is originally devised to speed up handcrafted detectors for better efficiency and has been extended by its follow-up works FAST-ER [RPD08] and ORB [Rub+11]. Among the early works, Šochman *et al.* is the first one to learn a whole process of interest point detector design, showing the possibility of emulating handcrafted detectors via learning. Richardson *et al.* [RO13] learn linear filters to compute detection response via convolution, where the lack of non-linearity makes its generalization across applications unclear. While the classical learning-based detectors show the potential of improving existing methods, as shown in [TM+08], they are shown to be less accurate than the purely handcrafted ones [HS+88; Mat+04; MS04].

Tilde [Ver+15] trains a linear regressor to predict a score for every patch in an image and then detect keypoints by thresholding the score map. It is supervised with a robust set of SIFT keypoints using classification loss. DetNet [LV16] proposes to train a CNN for detection based on feature covariant constraints. QuadNet [Sav+17] trains a CNN to produce keypoint response with an objective that enforces transformation-invariant ranking up to quantiles. MagicPoint [DMR17] predicts a dense keypoint heatmap from an image using a CNN, which is trained with synthetic corners obtained from rendered shapes. Kcnn [Di +18] trains a compact CNN to emulate the keypoint response of handcrafted detectors [ABD12; Low04]. [ZR18] shows a low ranking loss does not necessarily imply high repeatability of the detector, yet this is data-dependent. They then incorporate a peakness loss and increase receptive field in the network to enhance the repeatability. D2d[Tia+20] uses pre-trained L2Net [TFW17] to perform detections directly in its feature maps. Key.Net [Bar+19; LM22] combines handcrafted and learned CNN features at different scale levels. Instead of training with GT keypoint annotations, it follows DetNet [LV16] to learn keypoints with covariant constraints on a synthetic dataset.

3.3.2 Learned Descriptors

Deep learning-based keypoint descriptors are normally formulated as a metric learning problem, where the objective is to learn patch representation that is close to matching data and apart from non-matching data. With the advent of deep learning, convolution neural network is becoming a common option for extracting descriptors from image and image patches. A description network is typically trained by applying standard metric learning loss functions such as contrastive [CHL05] or triplet loss [SKP15; HA15; Ara+16] to the extracted descriptors. Data mining is also used to speed up the training and improve discriminativeness such as hard negative mining [Sim+15; Bal+16; Mis+17; Wei+18] and label mining via adding distractors [HLS18].

In the past years, learning-based descriptors have gained much attention and progress with advances in loss functions, network architectures, data mining strategies and training regularization. MatchNet [Han+15] and DeepCompare [ZK15] jointly learn patch representation via siamese deep CNNs and feature similarity comparison via shallow linear layers. To maintain efficient computation, MatchNet applies the patch description network and the metric network for matching separately during inference. DeepCompare shows that the learned descriptors can be more accurately compared using the common L2 distance. DeepDesc [Sim+15] also learns patch descriptors using a Siamese CNN and is trained on patch similarity, yet directly with a L2 distance metric instead of a learned metric. They further use hard positive and negative mining to learn discriminative features.

Later works [Bal+16; KCR16; Wei+18; Mis+17; Zha+17; Tia+19; Ebe+19; Wan+20b; Liu+19] leverage more advanced architecture, triplet loss and regularizers [Zha+17; Tia+19] for performance improvement. GLoss [KCR16] uses the global loss to enlarge the distance margin between positive and negative patch pairs. GOR [Zha+17] proposes a global orthogonal regularization loss that spreads out the learned descriptors to fully utilize the embedding space. HardNet [Mis+17] gains better performance via performing *hardest-within-batch* mining. GeoDesc [Luo+18] integrates geometry constraints from SfM during training. KSP [Wei+18] proposes a novel CNN subspace pooling method to learn descriptors that are more invariant to geometric transformations, leading to improved patch matching accuracy. SOSNet [Tia+19] shows that it is beneficial to incorporate a second-order similarity regularization loss for descriptor learning. ContextDesc [Luo+19] proposes to go beyond the local representation of a keypoint descriptor by aggregating simultaneously its visual contextual feature from image patches and geometrical contextual feature from keypoint locations. LogPolarDesc [Ebe+19] leverages polar representation to learn scale-invariant descriptors.

Instead of triplet loss, DOAP [HLS18] proposes to learn descriptors by optimizing on the matching average precision metric, which allows global descriptor comparison. To get rid of the need for expensive correspondence labels, CAPS [Wan+20b] supervises descriptors learning with an epipolar geometry-based loss that only requires ground truth camera poses and camera intrinsics to compute. They further show their method also outperforms most of the methods supervised with precise correspondences. Instead of extracting a descriptor independently for each image, CoAM [WEZ21] proposes to learn descriptors for a pair of visually overlapped images. Given an image pair, CoAM conditions the descriptor of one image with the features extracted from the other image via attention mechanism. The descriptors are trained with a hinge contrastive loss that minimizes the distance between two matched descriptors. During testing, descriptors are used to perform matching as usual. In this case, the descriptor of an image needs to be re-extracted for a new paired image, which makes it more expensive compared to other learned descriptors. While this character is similar to end-to-end correspondence networks, it differs from them in two aspects. A standard correspondence network performs an exhaustive matching step inside the network, which leads to expensive runtime and memory. Instead, it performs the matching outside the network to enable high-resolution dense descriptors and also avoids the expensive exhaustive matching during training by only backpropagating from a subset of correspondences.

3.3.3 Learned Joint Detection and Description

In the above paragraphs, we walked through both handcrafted and learning-based methods for only keypoint detection or description. In addition, we also introduced several handcrafted methods [Low04; BTG06; Rub+11; ABD12; AZ12] that do both duties, *i.e.*, detection and description, in two stages. In this section, we focus on another line of learned-based methods that opt for a single network to integrate detection and description in a single forward pass. The main intuition behind is that simultaneously optimizing the model parameters for these two inherently coupled tasks in an end-to-end manner leads to improved knowledge propagation and task cooperation. The existing methods along this line can be divided into three categories *detect-then-describe* [Yi+16; Ono+18], *detect-and-describe* [DMR18; Rev+19b; Dus+19; Luo+20] and *describe-then-detect* [Li+22].

Detect-then-describe. The first unified pipeline for local feature extraction is proposed in LIFT [Yi+16]. It combines the existing learning-based detector [Ver+15], orientation estimator and descriptor [Sim+15] in an end-to-end differentiable manner. While every component is differentiable, they show it is not possible to train the full pipeline from scratch. In the end, they gradually add components into training from only the detector to the full pipeline. Following a similar local feature pipeline as in LIFT, LFNet [Ono+18] adopts more advanced CNN architecture for more scale-invariant detection and can be trained end-to-end without the need for ground-truth keypoints generated by a handcrafted detector. Instead, they select keypoints from a score map using non-maximum suppression.

Detect-and-describe. Instead of connecting detection and description in sequential order, SuperPoint [DMR18] simultaneously outputs dense keypoints and descriptors. It extends its previously developed detector MagicPoint [DMR17] with another description branch to simultaneously generate a dense descriptor map. The detector and descriptor branches share the same CNN backbone for image feature extraction. While the detector outputs a heatmap at image resolution, every pixel-level descriptor is interpolated from the raw patch-level descriptors. To train SuperPoint end-to-end in a self-supervised manner, they leverage the pre-trained MagicPoint to generate pseudo ground-truth keypoint for real-world images and create ground-truth correspondences by image warping. It is shown to be very efficient in computation and has become one of the most popular local feature that is widely used in various applications and tasks. While the previous methods separate parameters for detection and description, D2Net [Dus+19] shares all parameters to obtain a common feature representation that is simultaneously optimized for both tasks. They show that the keypoints merge in the feature maps and can be extracted by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor. A similar conclusion has been drawn in other works [Tia+20; SAC19], where they directly use features learned on image classification. ASLFeat [Luo+20] further extends D2Net by fusing feature maps at multiple scales within a more sophisticated deformable CNN, leading to improved matching accuracy. Analogous to SuperPoint, R2D2 [Rev+19b] shares the backbone features and uses two separate branches for detection based on reliability and repeatability scores and dense description. By using L2Net [TFW17] as the backbone, it directly outputs feature maps at image resolution at the cost of slow computation. DISK [TFT20] proposes a probabilistic framework based on gradient policy for learning local features for image matching, which tries to keep the training and inference regimes close. They show their method outperforms other learned local features in image matching benchmark [Jin+21].

Describe-then-detect. PoSFeat [Li+22] reverses the traditional *detect-then-describe* pipeline to *describe-then-detect*, where it first trains a dense description network and then learns a detection network that predicts keypoint response from the dense descriptors. The idea of *describe-then-detect* keypoints has previously been presented in D2D [Tia+20]. However, D2D directly extracts keypoints without an extra detection network and focuses on detection and is combined with other descriptors for matching. Similar to CAPS [Wan+20b], PoSFeat is weakly supervised with camera poses. It outperforms other local features in image matching (HPatches [Bal+17]) and visual localization under night conditions (Aachen day and night [Sat+18]) while being competitive on SfM.

3.4 Robust Matching and Outlier Filtering

After obtaining a set of keypoints and their descriptors, the next stage is to establish keypoint correspondences. It is typically done by performing Nearest Neighbour (NN) search [ML14] based on the euclidean distance between every two descriptors. Afterward, simple filtering mechanisms such as Lowe's ratio test [Low04] and mutual consistency check [Dus+19; ZSL21] are usually applied to the initial set of matches to filter the unqualified ones.

Graph-based matching. Matching two sets of data points can also be considered as a graph matching problem. Given a set of keypoints extracted from an image, an attribute graph can be constructed where each keypoint is a node and an edge can be defined with incorporated pairwise constraints. In general, solving graph matching is normally formulated as Quadratic Assignment Problem (QAP) which is NP-hard and requires expensive and complex solvers. Therefore, despite being a long-standing topic, it is not widely applied to solve image matching in practice, as it is not affordable for the downstream applications that people focus on, such as visual localization, SfM and SLAM. While many relaxation strategies have been introduced in the literature to devise more efficient solvers for graph matching, we consider them as less related to this section and refer readers to other literature [Yan+16; Ma+21; SZF20] for detailed summary. In the following, we focus on visual descriptor-based keypoint association and mismatch filtering that have been widely applied to image matching.

Learn match filtering. Given an initial set of matches, the task of finding the set of inlier matches is typically solved jointly with the task of model estimation via RANSAC [CMK03; FB81; CWM05; Bra+17]. The final inlier set is picked as the maximum set of correspondences found by any of the randomly sampled model hypotheses within specific iterations. Some work proposes to learn a deep network to regress a probability score to directly weight matches inside a RANSAC loop [BR19b]. Another types of recent works [Yi+18; Sun+20; Zha+19a; Zha+19b; CYT19; Liu+21] propose to learn an extra filtering stage in data-driven manner to reject geometrically inconsistent matches before performing task-specific filtering inside RANSAC loops. It is normally formulated as a binary classification task where each match is assigned a probability score. The inlier matches are identified by setting a filtering threshold.

Learned matching. The seminal work SuperGlue [Sar+20] formulates descriptor matching as an optimal transport problem [PC+19] that can be efficiently solved with a differentiable iterative Sinkhorn solver [SK67; Cut13]. The motivation for learning a matching function is to learn the underlying structure of the problem, *e.g.*, not every point having a match and each point having at most one match, in a data-driven manner. SuperGlue leverages an attention-based Graph Neural Network (GNN) to explore and propagate the contextual information within each set and across two sets. The network is end-to-end trainable and can be trained with different types of local features. The combination of SuperPoint [DMR18] and SuperGlue has achieved the top rank on multiple matching and localization benchmarks since proposed and still being relatively competitive nowadays. Its follow-up work SGMNet [Che+21] significantly reduces the computation and memory cost of Superglue by constructing a sparsely connected seed graph neural network that operates on a set of seed matches, which are selected from nearest neighbor searched correspondences using non-maximum compression w.r.t. spatial coverage. Another similar recent work, ClusterGNN [Shi+22] directly removes redundant connectivity in the graph by progressively dividing keypoints into different sub-graphs, which leads to largely improved efficiency in computation meanwhile sets the state-of-the-art of learned matching function.

3.5 End-to-End Image Matching

In the above, we mentioned the methods that focus on tackling only one or two parts of a matching pipeline, while inheriting the remaining parts from other existing methods. While deep learning allows us to migrate from handcrafted designs to data-driven knowledge, there are still different levels of man-made assumptions involved in such a matching pipeline composed of separately devised components. One common assumption made to define a keypoint is that corners and conjunctions are resilient and invariant against various image transformations. In addition, while the ultimate application of the identified keypoints is to establish correspondences, they are not optimized directly on a matching objective. The above methods relying on classic keypoint detection and description perform *sparse-to-sparse* matching, since they first eliminate a set of pixels leading to *sparse* keypoints in each image. While the sparsity provides computation efficiency in the next matching stage, it might remove important information helpful to the matching step at the early stage, which is not recoverable anymore in the later stages. Therefore, recently a newly emerging trend is to learn a full or close to the full matching pipeline in end-to-end manner, which allows one to optimize directly on matching-based objectives. Those methods perform either *sparse-to-dense* [GBL19; GBL20; Tan+22; Jia+21; GLB21] or *dense-to-dense* [Hua+22; Roc+18; Li+20b; RAS20; Mao+22; Che+22a] matching as we will introduce in the following.

3.5.1 *Sparse-to-dense* Matching

Sparse-to-dense matching is firstly proposed by Germain *et al.* [GBL19]. where they use the VGG-based network trained on image retrieval losses to extract local features at a given point location. Given a pair of images and a set of identified keypoints in one of the images, S2DHM [GBL19] first extracts sparse descriptors at the keypoint locations for one image and dense descriptors per-pixel (called *hypercolumns*) for the other image. For each keypoint, its correspondence is obtained by exhaustively comparing the descriptor distance against all descriptors of the other images and keeping the one with the nearest descriptor distance. Its follow-up work S2DNet [GBL20] directly trains a network specifically for sparse-to-dense matching by maximizing the matching log-likelihood at ground-truth locations. They show by end-to-end training of the network on matching significantly improves the performance compared to a network trained on image retrieval.

Recently, *sparse-to-dense* has been reformulated in COTR, where given an image pair and a point of interest in one of the images, it outputs its matched point in the other image instead of a matching score. Instead of performing an explicit matching step to obtain the matching score, COTR considers a transformer [Vas+17] as a retrieval function that searches the correspondence of a query interest point from a context map. The map is constructed by concatenating CNN feature maps extracted from the image pair and appending each feature with its corresponding positional information in the image. One drawback of COTR is the significant computation requirement due to the use of a transformer, which also limits its accuracy since the image resolution is limited to 256 to avoid intractable runtime. ECO-TR [Tan+22] has been proposed recently to significantly reduce the runtime through hierarchical searching and more optimized implementation. Notice, one could also classify COTR and ECO-TR to *dense-to-dense* matching methods as they utilize all dense features from both images for matching. Yet, we put it in this category considering their need for pre-defined interest points.

3.5.2 *Dense-to-dense* Matching

While *sparse-to-dense* matching methods assume that the keypoints or query points are provided for one of the images in a pair, *dense-to-dense* methods, also called *detector-free* matching, directly forgo a keypoint detection stage. In this paradigm, image *keypoints* are not explicitly defined from the beginning but they are revealed directly in the identified matches. Fully end-to-end trainable *dense-to-dense* matching is firstly proposed in NCNet [Roc+18] where given a pair of images, dense descriptors are firstly extracted using a backbone network and then matched through a correlation layer [RAS17] to output a 4D correlation score map. It further applies several 4D convolution operations on the correlation tensor to obtain neighborhood consensus, which can be considered as a learned match filtering function. The final matches are identified by applying softmax from both matching directions and keeping the matches based on the highest probability and mutual consistency. Following this formulation, the later correspondence networks improve the matching accuracy by performing coarse-to-fine matching [Li+20b; Sun+21], designing a transformer-based [Vas+17] architecture to help to exploit contextual information [Sun+21]. Another set of methods focuses on making correspondence networks more computationally efficient using sparse convolution [RAS20]. Among the dense matching methods, LoFTR [Sun+21] achieves the best performance while being more efficient in runtime. Its follow-up work AspanFormer [Che+22a] proposes an advanced transformer architecture based on hierarchical attention showing improved performance in some cases. Concurrently, 3DG-STFM [Mao+22] shows a teacher-student learning strategy can be involved to improve the matching performance, where the teacher model is trained for 3D-3D matching with the same architecture as the student model for 2D-2D matching.

4 Visual Localization

In this chapter, we dive into visual localization. We start with the task definition in section 4.1 covering the task objective and the scene representation. Next, we classify the existing localization approaches based on their scene representation and involved techniques into five categories which include image retrieval-based localization (section 4.2), absolute pose regression (APR) (section 4.3), scene coordinate regression (SCR) (section 4.4), relative pose-based (RP-based) (section 4.5) and structure-based (section 4.6) localization. For each category, we discuss the key idea and motivation behind, the existing representative works, the current state and the remaining challenges.

4.1 Task Formulation

Objective and terminology. The goal of *visual localization* is to localize an agent that can be a human, a car, a robot or a drone, in the 3D environment based on some visual information provided by the target. The input visual information is commonly in the form of a single query image captured using a digital camera by the agent, so the task is also called *image-based localization*. Given a query image, a typical visual localization system localizes the agent by estimating the camera pose at which the query image has been captured w.r.t. the global coordinate system of the 3D environment. Therefore, the task is also termed as *camera localization* based on its camera pose output. As introduced in section 2.1) a camera pose involves a rotation and a translation, while some localization methods might only predict the positional information. In addition, the tasks are sometimes also called *camera re-localization* since we re-localize a new camera against a scene map that is also typically pre-built by localizing cameras. In this work, we stick to using *visual localization*.

Map representation. As the agent is localized w.r.t. a 3D environment, visual localization assumes that a map representation of the 3D environment has been built in advance. In general, there are three options to represent a map: i) a collection of images tagged with camera poses, ii) a sparse 3D point cloud and iii) a hybrid of i) and ii). The existing methods typically rely on the image database to obtain scalability in scene scale [Zho+20; Las+17; Sar+19], while leveraging a 3D model can lead to better localization accuracy [Sar+19; SLK16]. With the development of deep learning techniques, it becomes possible to encode the scene during training and then get rid of the explicit representation of the map during inference, such as in absolute pose regression (section 4.3) and scene coordinate regression (section 4.4).

Sparse map generation. A sparse scene map is commonly built by reconstruction techniques such as Structure-from-Motion (SfM) [Wu+11; SF16] or simultaneous localization and mapping (SLAM) [Dav+07; MT17]. SLAM reconstructs a 3D point cloud while taking images and localizing them in the target environment from scratch, while SfM builds a scene point cloud from a collection of images captured in the target scene. Both of the reconstruction techniques provide a sparse 3D point cloud, as well as a database of images with their localized camera poses, where these images are called *reference images*. Each 3D point in the 3D model has been triangulated from two or more image local features (*c.f.* section 3.2 and section 3.3) such as SIFT [Low04]. In this way, each 3D point cloud is associated with one or multiple local image descriptors. Compared to a collection of images, a sparse 3D point cloud is more *compact* and more *informative* representation, since points with redundant and ambiguous local features are already eliminated during the reconstruction [Li+12]. It's worth noting that a reconstructed map algorithm only generates pseudo ground truth which also contains different levels of errors. Recent study shows the choice of algorithm to generate reference map can influence the ranking of localization methods on the same dataset [Bra+21].

Dense map generation. RGB depth SLAM is often used to build map for indoor scenes where 3D structure is densely reconstructed from RGB-D images, *i.e.*, images with RGB and depth values [New+11; Sho+13; Val+16; Bra+21; Dai+17]. Different from SfM which works with loosely captured scene images, depth SLAM requires video sequence where images are captured at high frame rate. Another case is to directly scan the whole scene and then register the images to a floor plan as was done for InLoc [Tai+18; WF17] with handcrafted alignment. Some recent work also proposes to use a dense scene representation in the form of neural radiance fields [Yen+21; Mil+21] or meshes [PKS22].

4.2 Image Retrieval-based Localization

End-to-end. Traditionally, visual localization can be cast into an image retrieval (IR) task [SH+04; Hum+22; SBS07; KSP10; CS13; Tor+15; BMC22; Wan+22], when using a database of geo-tagged images as the scene representation. As introduced in section 2.3, given a query image, image retrieval searches the nearest neighbors that observe content similar to the query from a reference database. Then the query location can be directly approximated by the location of the nearest reference image. In the context of *visual place recognition* [MC21], the reference tags are not constrained to precise camera poses, but also can be GPS coordinates [HE08; Pra+22; WKP16]. In our previous work [Sat+19], we show that compared to other methods that output a precise camera pose, end-to-end localization via IR has the coarsest localization accuracy yet but generalizes well across datasets.

Coarse-to-fine. While end-to-end localization via image retrieval is less accurate, it allows one to efficiently provides candidate regions. This character is complementary to feature-based methods that are more accurate but expensive to compute on large-scale. Therefore, image retrieval is naturally used as a coarse searching step to first reduce the localization space, followed by a fine localization step that is performed only in the candidate space. Such a coarse-to-fine localization paradigm has been commonly adopted since the early days until nowadays by structure-based localization [Irs+09; Sar+19; Zho+22; Tai+18] (*c.f.* section 4.6) and relative pose-based localization [ZK06;

Zho+20; Las+17] (*c.f.* section 4.5) methods. Recently the hierarchical structured-based localization [Sar+19] has achieved the state-of-the-art performance in the challenging long-term localization benchmark [Sat+18; Tof+20], either being more accurate or scaling much better to larger scenes compared to other methods. Among the existing retrieval methods, the VLAD-based ones such as NetVLAD [Ara+16] and DenseVLAD [Ara+16] are the most widely applied ones to visual localization.

4.3 Absolute Pose Regression

Directly regressing a 6-DoF camera pose from a query image was firstly proposed by Kendall *et al.* [KGC15] as a lightweight and real-time localization solution. The vanilla PoseNet [KGC15] firstly uses a convolutional backbone, *e.g.*, GoogleNet [Sze+15], to encode a query image as a global feature embedding, which is then fed into a series of fully-connected layers to regress its camera pose. To train a PoseNet for the target scene, one assumes a training database of images with ground truth (gt) camera poses are known. The training loss is typically computed as a weighted combination of the translation residual between the predicted and gt translations and the orientation residual between the predicted and gt orientations. A bunch of latter works build upon the vanilla PoseNet [KGC15] formulation and gradually improves its performance by designing more sophisticated architecture [NB17; Mel+17a; Wan+20a; SFK21], reasoning about the uncertainty of the estimations [KC16], introducing geometric constraints [KC17; Bra+18; VRB18; RVB18] in losses, exploring temporal or geometric cues when using image sequences [Wal+17; Cla+17; VRB18; RVB18] or multiple non-consecutive images [Xue+20] as inputs.

Novel view synthesis-aided APR. Recently, dense scene representation in the form of mesh or neural radiance fields (Nerf) [Mil+21; Mar+21; Bar+22] are used to improve absolute pose regression (APR). Novel views are densely rendered from a pre-trained nerf [Mor+22] or a textured mesh [Sat+19] to enlarge the reference database used for training. Direct-PoseNet [CWP21] enforces photometric consistency between the input image and the rendered view using its predicted poses during training. This is later extended by DFNet [Che+22b] to compute the consistency in feature space.

Multi-scene APR. While having the merit of simple end-to-end formulation and constant runtime regardless of scene size, APR methods are scene-dependent by design as they are trained to be dependent on the choice of the global world coordinate. To empower an APR model to work under multiple scenes, MSPN [Bla+20] learns shared feature extraction across multiple scenes and separates regression components for each scene without loss in accuracy compared to training one full network per scene. The recent work [SFK21] manages to train a single full APR network across multiple scenes by leveraging transformers to learn general features for localization while embedding multiple scenes in parallel.

4.3.1 Challenges

Despite the years of development in absolute pose regression, this branch of methods are still relatively less accurate compared to other approaches that explicitly compute camera poses from keypoint correspondences between a query and the scene geometry [BR21; Sar+19; Li+20a]. In order to investigate the inner workings of APR methods and to understand the reason behind their lower performance compared to structure-based methods, we proposed a theoretical model in our previous work [Sat+19] where we interpret the weight parameters of the last linear layer that regresses image camera poses as a set of base poses, while a predicted test pose is a linear combination of the base poses. Those base poses can be viewed as the representation of the scene that an APR network learned from training poses. This theory implies the sufficient condition for a testing image to be successfully localized is its pose can be actually represented by a linear combination of the base poses. However, in real-world applications the training poses might be sparse or simply distributed distant from the testing cameras. To avoid those cases one would need to densely sample more training cameras to improve viewpoint coverage, however, it leads to more expensive data collection as well as longer training time in general. In contrast, the structure-based localization methods are less constrained by the scene coverage extent. As long as there exists sufficient scene structure visible to both the query and some reference images, it is possible to localize the query image.

4.4 Scene Coordinate Regression

The original scene coordinate regression (SCR) framework for camera localization was proposed by Shotton *et al.* [Sho+13] where instead of performing explicit 2D-3D sparse feature matching, it regresses the matched 3D scene coordinate for each pixel in the query image from its local patch feature using a random forest. Its follow-up works improve the localization accuracy by using multiple camera pose predictors [Guz+14] or modeling regression uncertainties [Val+15].

DSAC-based SCR. Unlike the previous work using random forests [Sho+13; Guz+14; Val+15; Bra+16; Cav+17; Cav+19b] and handcrafted local patch features, Brachmann *et al.* trained a CNN to densely regress scene coordinates from raw RGBs, meaning feature extraction and feature-to-scene matching are jointly learned from training data. Such modification significantly improves over prior works on 7 Scenes dataset [Sho+13]. They further proposed DSAC, a differentiable approximation of RANSAC [FB81] to enable end-to-end training on pose estimation yet requiring initialization from pre-trained components for stability. Its follow-up work DSAC++ [BR18] relaxed the need for depth map for scene coordinate regression training with a depth prior and improved the stability of end-to-end training by using a soft inlier count instead of a learnable CNN for hypothesis scoring in DSAC, which led to improved localization accuracy for both indoor and outdoor scenes. DSAC* [BR21] further simplified the training procedure and extended to leverage sparse 3D scene model during training or dense depth maps for both training and inference leading to increased accuracy.

Large-scale SCR. While scene coordinate regression approaches achieve highly accurate localization performance in benchmark with small to middle scale scenes, *e.g.*, 7 Scenes [Sho+13], 12 Scenes [Val+16] and Cambridge Landmarks [KGC15], directly training them on larger scenes led to much worse performance [Bra+17; BR18; BR19a]. To relieve this issue, Brachmann *et al.* proposed ESAC [BR19a] where they train a collection of experts with each expert responsible for regressing scene coordinates for a sub-area in a larger environment and an additional gating network for identifying which expert to use for an input image. A recent follow-up work proposed a hierarchical classification and regression network [Li+20a] that leads to improved accuracy on indoor scenarios. In addition, they show they can significantly outperform ESAC on the challenging large-scale Aachen Day and Night [Sat+12; Sat+18] dataset by coarse-to-fine classification without regression with sparse feature instead of the raw image as inputs. Despite the improvement made in the recent work [BR19a; Li+20a], scene coordinate regression / classification methods are much less accurate than the state-of-the-art structure-based localization on large-scale datasets such as in Aachen Day and Night.

Online-adaptation to new scenes. In parallel with advancing the accuracy and run-time efficiency of SCR methods, some other works [Cav+17; Cav+19b; Cav+19a] extend their application for interactive SLAM. The idea is to leverage a localizer trained offline on one or some scenes (not the target scene) for online re-localization on a new scene. More specifically, online adaption is the process of training the offline localizer on new data obtained by a camera tracker in the target scene. Afterwards, the adapted localizer can be used to recover the camera pose when the camera tracking fails.

Scene agnostic SCR. In order to overcome the scene-dependent constraint for SCR, SANet [Yan+19] leverages the 3D model of the scene. Compared to structure-based methods, the main difference is they do not perform explicit 2D-3D matching, instead they explore feature correlation using MLP and then regress scene coordinates for the query image. While they show the model generalizes to some extent, their performance is still lower than explicit matching-based methods considering they now also need 3D model for inference.

Relation to APR. Besides camera poses regression, scene coordinate regression is another popular regression-based method for visual localization that has benefited largely from deep learning in the past years. Different from absolute pose regression which learns the whole localization pipeline in an end-to-end manner, scene coordinate regression only learns to establish 2D-3D correspondences and still requires another pose estimation stage as in structure-based localization.

4.5 Relative Pose-based Localization

In the previous section 4.3 and section 4.4, we have introduced methods that directly predict camera poses from query images without the need for an explicit map during the inference. Instead of learning to encode the scene representation inside the model parameters, relative-based localization relies on a database of reference images depicting the scene to work.

Pipeline. A typical relative-based localization pipeline consists of three stages: 1) image retrieval is used to identify a set of images that potentially depict the same part of the scene as the query image, (2) for each retrieved image, its relative camera pose w.r.t. the query is computed, and 3) relative poses from all retrieval images are used to estimate the final absolute camera pose of the query. In recent years, deep learning has been widely involved to improve both steps 1) image pair construction and step 2) relative pose estimation. As image retrieval is the core technique used in step 1), we refer readers to section 4.2 for detailed description of how it works and how deep learning has been applied to advance it. For relative pose estimation, there are mainly the regression-based methods and the matching-based ones. The former methods directly regress a relative pose from a pair of images with visual overlapping, being purely learning-based. The latter follows the classical keypoint matching paradigm to establish image correspondences which are then used to compute relative poses using minimal solvers for relative pose estimation such as 5-point solvers [LH06; Nis04] for known camera intrinsics and 6-point solvers[Kuk+17; HL12; Ste+08] for unknown focal length.

From handcrafted to learning-based. The first relative pose-based localization [RC04] system was matching-based. Given an available database of rectified images of building facades, Robertson & Cipolla [RC04] perform wide-baseline matching between a rectified query image and each database image. They then estimate the relative transformation, *i.e.*, a scale and a translation, between the best matched image pair, from which they finally recover the pose of the query given the pose label of the database image. The latter work [ZK06] generalizes this prototype by eliminating the requirement that each image needs to be dominant with a building facade. In addition, they use more viewpoint and scale invariant SIFT [Low04] feature to perform wide-baseline matching and then estimate relative motions between the query to its two best retrieved database images, which finally are used to triangulate the global position of the query given the two GPS coordinates of the database images. However, in the following many years, relative pose estimation was not much investigated in the context of visual localization.

Until a few years ago, more and more researchers start to explore the potential of using deep learning to tackle various geometric computer vision tasks in an end-to-end manner, including camera localization [KGC15], visual odometry [KM15] and homography estimation [DMR16]. Following the trend of deep regressing geometric transformations, Melekhov *et al.* [Mel+17b] regress relative camera poses from image pairs using a siamese network with a weighted sum of L2 loss on normalized predicted and ground truth relative translations and quaternions. Laskar *et al.* [Las+17] is the first one that evaluates relative pose regression (RPR) for the task of camera re-localization using a similar pipeline to the one proposed in [ZK06] except for the RPR step. In addition, they show the possibility to train a single RPR network on a set of scenes indicating that RPR-based localization has the potential to be more general and scalable than the scene-dependent APR. Balntas *et al.* [BLP18] train a network for joint image retrieval and relative pose regression using a frustum-overlap-based loss. Instead of triangulating the query position, several other work [BLP18; ELJ18; ABI21] learn metric relative poses, *i.e.*, rotation and scaled translation, from which query poses can be directly recovered. RelocNet [BLP18] shows that compared to training on the target dataset (7 Scenes [Sho+13]), training their model on another similar dataset (ScanNet [Dai+17]) leads to an evident decrease in performance, suggesting that the method has certain but limited generalization capability across datasets. AnchorNet [SVJ18] exhaustively predicts xy-offset between a query and a selected set of reference images called anchor points. For the other 4-DoF, it regresses z-axis and orientation globally as in APR. While such a combination of APR and RPR leads to performance improvement compared to its previous APR and RPR methods, it needs to be trained per-scene.

4.5.1 Towards Generalized and Accurate RP-based Localization

Compared to relative pose estimation via explicit feature matching, the above mentioned RPR formulation does not enforce the intrinsic structure of the problem. Instead, the underlying belief is that implicit feature matching is performed within regression and it is supposed to be competitive or even better than enforcing explicit feature matching in some aspects such as simplicity, accuracy or generalization. If not, this also raises a question that *what is the proper way to leverage learning for relative pose estimation?* To explore the answer to this important question, we proposed EssNet [Zho+20] (chapter 5), an essential matrix-based framework that supports fair comparison between various relative pose estimation methods for visual localization. In our work, we compare three different variants for computing essential matrices, ranging from purely hand-crafted to purely data-driven. Through extensive experiments, we found purely data-driven RPR does not generalize well across datasets, *e.g.*, from indoor to outdoor, which is not an issue for matching-based variants.

A recent work ExReNet [WDT21] shows that it is possible and even beneficial to generalize from unrelated data without retraining. Interestingly, they show significant improvement in accuracy and generalization compared to previously proposed RPR models, including EssNet [Zho+20]. This is achieved by enforcing a matching operation via hierarchical correlation layers supervised with GT correspondences as well as increasing the pose regression layers before pooling. They verified their model can train on a synthetic indoor dataset SUNCG [Son+17] and generalize to the indoor 7 Scenes [Sho+13] dataset. They further show leveraging learned translation scale and uncertainty information in pose triangulation process significantly improves the translation error.

Another recent work [Arn+22] proposed to tackle *map-free* visual localization where the goal is to localize the query against a single reference image that represents the scene. The main motivation behind is to enable instant AR capabilities at new locations by getting rid of a time-consuming mapping stage. They extensively evaluated the performance of multiple localization methods against different types of scene representations, where they gradually restrict the representation to only a single reference frame. When assuming using all mapping frames (reference images) in 7 Scenes [Sho+13], as expected, structure-based methods (section 4.6) relying on a 3D model achieve the best accuracy. The second best performing method applies the state-of-the-art image matching to estimate relative camera pose and then triangulates query pose. When assuming only 10 scene frames, the relative pose estimation with triangulation becomes the best performing as 3D model is not available in this case.

In the extreme case of a single reference frame where triangulation is not possible, the most accurate method relies on learned depth to recover the translation scale. However, in any of the scenarios, one can see that relative pose regression with translation scale prediction is always the least accurate method, while relative pose estimated via image matching leads to significantly more accurate and robust performance. This observation is consistent with what we concluded from EssNet [Zho+20] that pure relative regression suffers from limited generalization issue and accuracy.

4.6 Structure-based Localization

In this section, we introduce structure-based localization which is currently the most reliable and accurate localization for long-term localization benchmark[2] [Sat+18]. They are also the main branch of methods that generalize the best across both indoor and outdoor datasets. In the following section 4.6.1, we first give an overview of the structure-based localization approaches and their relations to some of the other approaches that we have mentioned above. Next in section 4.6.2, we look into the heart of the structure-based localization, *i.e.*, how to establish 2D-3D correspondences in different paradigms. Finally, we discuss the open challenges facing the structure-based localization in the long run and draw conclusions from there.

4.6.1 Overview

Pipeline. Different from the other methods mentioned in the previous sections, structure-based localization assumes the availability of a pre-built 3D map (*c.f.* section 4.1) of the scene during inference. Given a query image and a 3D map, a structure-based localization pipeline can be summarized into two steps: (i) establishing point correspondences between 2D pixels in a query image and 3D scene points and (ii) computing the camera pose of the query using PnP solvers [Gao+03; KSS11; Lar+19]. Based on whether image retrieval is leveraged in step (i), structure-based methods can be divided into *direct* and *indirect* methods, which we detail in section 4.6.2. Once we obtain 2D-3D correspondences, we apply Perspective-N-Point (PnP) solvers [KR17; KSS11; Gao+03] to compute the final pose. With the modern data-driven image retrieval [Ara+16; RTC18] and image matching [DMR18; Sar+19; Sun+21; ZSL21] techniques, *indirect* methods achieves the current state-of-the-art localization performance.

Relation to scene coordinate regression. Scene coordinate regression (SCR) is closely related to structure-based localization but they differ from each other in several aspects. While SCR methods also establish 2D-3D correspondences for pose estimation, their matching step is implicitly performed via regression. Moreover, there exists no explicit representation of scene 3D structure in the inputs, instead, the network is the one predicting the 3D coordinates that correspond to the image pixels. As such, we consider the SCR formulation to be relatively different from the methods we want to discuss in this

[2] Long-term localization benchmark: https://www.visuallocalization.net/benchmark

section. Therefore, to avoid abuse of the terminology in this thesis we use *structure-based* only for methods that explicitly perform a matching step against a 3D scene model. We leave SCR itself as another type of approaches and detail it in section 4.4.

Relation to relative pose-based approaches. The relative pose-based (RP-based) localization methods are more similar to *indirect* structure-based localization when the relative pose is estimated via visual feature matching [ZK06; Zho+20; Arn+22]. For both types of methods, 2D-2D pixel correspondences are firstly established, however, structure-based methods use them to further obtain 2D-3D correspondences given the 3D scene model, leading to a different pose estimation stage. While RP-based methods are more lightweight in terms of the scene representation requirement, recent study shows they are less accurate than the structure-based localization even with the same feature matching stage [Arn+22].

4.6.2 Establishing 2D-3D Matches

The existing 2D-3D matching methods for structure-based localization can be classified into *direct* [SLL02; CS14; CN12; Gep+19; Li+12; LSH10; Lyn+15; SLK16; LLD17; Svä+16; ZSP15; Che+19] and *indirect* [Irs+09; Sar+18; Sar+19; Tai+18; Zho+22; PKS22] approaches. The main difference between the two categories is whether a method uses image retrieval [SZ03; NS06; Ara+16; Tor+15]. Sometimes we also call *indirect* methods *hierarchical* localization [Sar+18; Sar+19] as they firstly perform coarse localization via image retrieval and then fine localization via feature matching.

Direct 2D-3D matching. *Direct* 2D-3D matching methods for localization establish 2D-3D correspondences by globally comparing visual features, *i.e.*, keypoints and descriptors, extracted from a query image with 3D scene points (associated visual descriptors). In the early work, Se *et al.* [SLL02] show the feasibility of directly matching query SIFT [Low99] features against a scene model build via SLAM [SLL01] for mobile robot localization in a $10 \times 10m^2$ area. They also devised a RANSAC algorithm to efficiently filter the outlier matches. Later, the advances in Structure-from-Motion (SfM) techniques [SSS06; Fra+10] make it possible to reconstruct 3D geometry of landmarks from large collections of Internet photos, which motivates the research in larger city-scale (landmark-scale) localization. Following [Irs+09], Li *et al.* [LSH10; SLK11] examine direct 2D-3D matching with prioritized searching from both directions, *i.e.*, 2D-to-3D and 3D-to-2D, for globally localizing against large-scale outdoor scene models reconstructed by SfM. However, as the scene scale grows, localization methods based on direct 2D-3D matching are faced with several challenges. A large fraction of the 3D point features can become visually ambiguous due to repetitive structures, which leads to more wrong matches. In addition, the memory footprint and computational requirements of performing matching become prohibitive for many real-world applications. To tackle those challenges, a lot of research effort has been devoted to making matching more efficient [Lim+12; Li+12; CN12; SLK11; DS14; SLK16; LLD17; Che+19] as well as more accurate by devising more robust outlier filtering schemes [Svä+16; Aig+19; Aig+21; ZSP15; Sva+14; Li+12].

Indirect 2D-3D matching. With enormous progress made in image retrieval for place recognition and SfM for accurate scene geometry reconstruction, Irschara *et al.* [Irs+09] combines the two methods into a single pipeline for efficient large-scale localization. Specifically, they first use a vocabulary tree to retrieve top-ranked relevant reference images w.r.t. a query image [NS06], then perform tentative feature matching only between the query and each of the retrieved images and finally estimate pose from the matches using a PnP solver. They show such *indirect* structure-based localization delivers real-time performance against large 3D models obtained from SfM. Following this pipeline, Sarlin *et al.* [Sar+18] leverages a deep neural network for image retrieval and hand-crafted local features for matching, which enables highly accurate real-time localization on a mobile platform. Similarly, InLoc [Tai+18] shows that deep features are helpful to address challenges of long-term indoor localization which contain lots of large textureless areas, symmetric and repetitive elements and dynamic objects. Using NetVLAD [Ara+16] for image retrieval, multi-scale VGG [SZ14] features for dense matching and a novel pose verification step based on virtual view synthesis, they achieve significant improvements compared to state-of-the-art localization methods for large-scale indoor localization. Sarlin *et al.* [Sar+19] draws a consistent conclusion by jointly training a network for both global and local feature matching for large-scale outdoor localization. Recent research [ZSL21; Sun+21] shows that inserting the latest data-driven feature matching [DMR18; Sar+20; Sun+21] inside such a hierarchical localization (HLoc[3]) framework leads to the state-of-the-art performance in long-term localization benchmark for both indoor and outdoor scenes [Sat+18].

4.6.3 Challenges

As presented in section 4.6.2, both *direct* and *indirect* structure-based localization rely on visual descriptors to establish 2D-3D correspondences. While structure-based methods based on visual descriptor matching are highly accurate and robust, they face multiple challenges from storage efficiency, privacy perseverance and descriptor maintenance, blocking the development of large-scale localization for real-life applications.

Storage / memory efficiency. Storing per-point visual descriptors for matching makes a localization system demanding in storage when considering keeping those data on a server, since a city-scale 3D model with descriptors can easily rise up to the magnitude of TBs [Zho+22]. In the case of *direct* structure-based localization on mobile devices, loading the whole 3D model with associated descriptors into memory is not even possible for larger scenes, making this type of methods not feasible.

To keep memory footprint manageable, several works propose to directly compress the scene model while maintaining the localization accuracy as much as possible. The compression is typically done by keeping a more representative and compact subset of the 3D points [Soo+13; LSH10; Cam+19; CS14; Lyn+15; MSF20; Cha+22] and quantizing [Cam+19; Lyn+15; Sat+15] the descriptors associated with the 3D points. The selection of a more representative subset of 3D points is usually modeled as a K-Cover problem [Soo+13; LSH10; Cam+19; CS14; Lyn+15] such that each database

[3] HLoc official code release: https://github.com/cvg/Hierarchical-Localization

image observes at least K points from the selected subset. In the recent work, Chang *et al.* [Cha+22] propose to sparsify a SfM in a data-driven manner, where they train a graph neural network to predict point importance scores based on the local appearance and the spatial context in the map graph built from the SfM model.

Privacy perseverance. In the past years, feature inversion techniques have been widely studied originally for visualization and interpretation purposes, where a network is learned to reconstruct an image from the handcrafted [KH14; dAn+13; Von+13] or deep features [MV15; DB16] extracted from that image. Recently Pittaluga *et al.* [Pit+19] show their learned inversion network can remarkably reveal scene contents of a highly sparse SfM point cloud from its associated descriptors.

The existence of those powerful inversion techniques has raised privacy concerns in several related tasks such as visual localization [CKS21], SfM [Gep+20] and SLAM [Gep+21]. To defend against those inversion attack on a 3D scene map, Speciale *et al.* [CKS21] lift the map representation from a 3D point cloud to a 3D line cloud to obfuscate the geometry. However, its follow-up work [CKS21] shows that inversion attacks can still be successfully conducted to such a line cloud map. In fact, for modern localization systems following a server-client model, the inversion attack not only can happen on the server storing the 3D map, but it can also happen to query descriptors exposed during the transmission from client to server side. To mitigate this scenario, privacy-preserving descriptors [Ng+22; Dus+21b] have been recently proposed.

Descriptor maintenance. The continuous development of modern visual descriptors has significantly contributed to the increasing localization performance. To allow the existing localization system to use more advanced descriptors, the 3D scene model needs to first be updated such that the 3D points are also associated with the new type of descriptors. However, this process typically requires firstly extracting local features from all database images, exhaustively matching between every pair of images, and finally re-triangulate the 3D points. As the bigger the scene size is the more expensive this process costs, it becomes prohibitive to perform such an update for localization systems deployed at city-scale.

Other than the above mentioned issue, Dusmanu *et al.* [Dus+21a] also points out another challenging scenario where different feature extraction algorithms are running on different devices, leading to incompatible matching across devices. To address those challenges, they jointly train a set of encoder-decoder networks for descriptor conversion, where each network is responsible for one descriptor algorithm. During inference, they convert the 3D map descriptors to the target query descriptor type to perform matching. However, their experiments demonstrate such conversion decreases the quality of modern learning-based descriptors in general. While this method avoids the descriptor update, it either needs to constantly convert on-the-fly which makes the localization slower, or keep multiple versions of 3D model on the server which requires huge amount of storage for larger scenes as mentioned previously. Furthermore, retraining all of the encoder-decoder networks jointly every time a new type of descriptor emerges is another type of continuous maintenance effort.

A potential solution via geometric-based matching. As the three mentioned challenges are all related to the use of visual descriptors to establish 2D-3D keypoint correspondences, we wonder *whether we can localize an image without relying on visual descriptors ?* This question motivated our work GoMatch to be presented in chapter 7. Compared to other methods that aim to tackle one aspect of those challenges, it explores an orthogonal path to address these challenges by performing geometric-based matching which gets rid of the need for visual descriptors.

Part II

Publications

5 To Learn or Not to Learn: Visual Localization from Essential Matrices

5.1 Summary

A recent trend in leveraging deep learning for visual localization involves directly regressing relative poses from image pairs. Unlike absolute pose estimation, which is specific to a particular scene, predicting the relative pose between images is a task that should generalize to unseen scenes.

By representing the scene as a database of reference images with known poses, one can localize a query image by estimating its relative pose with respect to k reference images. Relative-pose-based methods, in contrast to structure-based methods, do not require storing a 3D scene model, which can be prohibitively expensive for large scenes. However, they currently exhibit lower accuracy compared to state-of-the-art structure-based approaches.

In this work, we aim to address two key questions regarding relative pose regression techniques: (i) why they currently exhibit lower accuracy than explicit matching-based methods, and (ii) whether they can generalize to new scenes. To this end, we propose a novel and versatile framework for relative-pose-based visual localization.

Our pipeline comprises three stages: We employ image retrieval to identify a set of images that potentially depict the same part of the scene as the query image. For each retrieved image, we compute the essential matrix encoding its relative pose with respect to the query image. Leveraging the known absolute poses of the retrieved images and the essential matrices, we estimate the absolute pose of the query. Our framework is agnostic to the method used for estimating relative poses in the form of essential matrices. This flexibility enables us to analyze the impact of employing machine learning in various ways for relative pose estimation on localization accuracy. We compare three approaches: (a) a classical method based on SIFT features [41], (b) direct regression of an essential matrix using a novel CNN-based approach proposed in this paper, and (c) a hybrid approach that employs learned feature matching instead of SIFT.

In our detailed experiments, we make the following observations: 1) Despite its simplicity, our SIFT-based approach proves to be competitive with significantly more complex state-of-the-art methods, thereby validating our framework. 2) Our regression-based approach, while surpassing previous work, still lags significantly behind the SIFT-based variant. Furthermore, it does not generalize to unseen scenes due to limitations in the ability of its regression layers to learn the fundamental concepts underlying relative pose estimation. 3) While the regression layer plays a key role in the inaccurate pose estimates of relative pose regression-based methods, it is not the sole aspect in need of improvement. Employing features learned by such methods in our hybrid approach also yields less accurate results compared to the SIFT-based approach. In addition to introducing a novel localization framework, this paper provides valuable insights for future endeavors towards achieving truly generalizable learning-based visual localization.

5.2 Author Contributions

The author of this dissertation significantly contributed to

- developing the main concepts
- implementing the algorithm
- evaluating the numerical experiments
- writing the paper

5.3 Preprint

Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 3319–3326

Following the IEEE reuse permissions, we include the *accepted* version of the publication. The published version is available under: https://doi.org/10.1109/ICRA40945.2020.9196607.

To Learn or Not to Learn: Visual Localization from Essential Matrices

Qunjie Zhou[1], Torsten Sattler[2], Marc Pollefeys[3,4], Laura Leal-Taixé[1]

***Abstract*— Visual localization is the problem of estimating a camera within a scene and a key technology for autonomous robots. State-of-the-art approaches for accurate visual localization use scene-specific representations, resulting in the overhead of constructing these models when applying the techniques to new scenes. Recently, learned approaches based on relative pose estimation have been proposed, carrying the promise of easily adapting to new scenes. However, they are currently significantly less accurate than state-of-the-art approaches. In this paper, we are interested in analyzing this behavior. To this end, we propose a novel framework for visual localization from relative poses. Using a classical feature-based approach within this framework, we show state-of-the-art performance. Replacing the classical approach with learned alternatives at various levels, we then identify the reasons for why deep learned approaches do not perform well. Based on our analysis, we make recommendations for future work.**

I. INTRODUCTION

Given a query image, the goal of visual localization algorithms is to estimate its camera pose, *i.e.*, the position and orientation from which the photo was taken. Visual localization is a fundamental step in the perception system of robots, *e.g.*, autonomous vehicles [40], [42], and a core technology for Augmented Reality applications [3], [12].

Current approaches to visual localization that achieve state-of-the-art pose accuracy are based on 3D information [6], [13], [45], [60], [63], [68], [73], [74]. They first establish 2D-3D matches between 2D pixel positions in a query image and 3D points in the scene. The resulting correspondences are then used to estimate the camera pose [21], [34]. The 3D scene geometry can either be represented explicitly through a 3D point cloud or implicitly via the weights of a convolutional neural network (CNN). Both types of representations are scene-specific, *i.e.*, they need to be build per scene and do not generalize to unseen scenes.

A more flexible scene representation models a scene through a set of database images with associated camera poses [65]. Building such a scene representation is trivial as it amounts to adding posed images to a database. The pose of the query image can then either be approximated by the pose of the most similar database image(s) [1], [75], [76], [83], identified through image retrieval [50], [71], or computed more accurately [65], [73], [86]. Multiple methods based on deep learning have been proposed for estimating the pose of the query relative to the database images [4], [35], [44], [77], [87] rather than to compute it explicitly from feature matches [86]. However, such approaches do not consistently perform better than a simple retrieval approach that only approximates the query pose [66].

Visual localization approaches based on relative poses have desirable properties, namely simplicity and flexibility of the scene representation [65] and easy adaption to new scenes, compared to 3D-based approaches. Also, leveraging modern machine learning techniques for relative pose estimation seems natural. This leads to the question why learning-based approaches do not perform well in this setting.

The goal of this paper is to analyze the impact of machine learning on relative pose-based localization approaches. To this end, we propose a novel and generic framework for visual localization that uses essential matrices inside a novel RANSAC scheme to recover absolute poses. Our framework is agnostic to the way the essential matrices are estimated. We thus use it to analyze the impact of employing machine learning in various ways: We compare (a) a classical approach based on SIFT features [41] to (b) directly regressing an essential matrix (using a novel CNN-based approach proposed in this paper) and (c) a hybrid approach that uses learned feature matching instead of SIFT. Through detailed experiments, we show that: 1) Our SIFT-based approach (a), despite its simplicity, is competitive with respect to significantly more complex state-of-the-art approaches [6], [63], [73], thus validating our framework. 2) Our regression-based approach (b), although outperforming previous work, is still significantly worse than the SIFT-based variant. Also, it does not generalize to unseen scenes due to the inability of its regression layers to learn the general concepts underlying relative pose estimation. 3) While the regression layer is mainly responsible for the inaccurate pose estimates of relative pose regression-based methods, it is not the only part that needs improvements. Rather, using features learned by such methods in our hybrid approach (c) also leads to less accurate results compared to (a). Besides proposing a novel localization framework, this paper thus also contributes important insights into future work towards truly generalizable learning-based visual localization.

II. RELATED WORK

Feature-based localization. Feature-based approaches to visual localization can be classified into *direct* [11], [14], [22], [37], [38], [42], [63], [72], [85] and *indirect* [2], [10], [28], [60], [61], [75], [84], [86] approaches. The former follow the strategy outlined above and obtain 2D-3D matches by directly comparing feature descriptors extracted from a query image with 3D points in the SfM model. While

This research was partially funded by the Humboldt Foundation through the Sofja Kovalevskaya Award.

[1]Technical University of Munich
[2]Chalmers University of Technology
[3]Department of Computer Science, ETH Zürich
[4]Microsoft Zürich

producing accurate camera pose estimates, their scalability to larger scenes is limited, partially due to memory consumption and partially due to arising ambiguities [37]. The former can be addressed by model compression [9], [11], [38], [42], [62] at the price of fewer localized images [11], [42].

Indirect approaches first perform an image retrieval step [29], [50], [71] against the database images used to build the SfM model. An accurate pose estimate can then be obtained by descriptor matching against the points visible in the top-retrieved images [28], [62], [73], which can be loaded from disc on demand. The retrieval step can be done very efficiently using compact image-level descriptors [1], [52], [75]. It is not strictly necessary to store a 3D scene representation for accurate pose estimation: Given the known poses of the database images, it is possible to compute the query pose via computing a local SfM model online [65] or by triangulating the position of the query image from relative poses w.r.t. the database images [86]. While [86] is limited to using only two database images, we propose a more general RANSAC-based approach to use more database images.

Learning for visual localization. Retrieval methods [1], [10], [23], [51] have benefitted greatly from deep learning. For 3D structure-based localization, several works have proposed to directly learn the 2D-3D matching function [5], [6], [13], [19], [24], [70], [79], [81]. Their main drawback, besides scaling to larger scenes [6], [69], is that they need to be trained specifically per scene. Recent work has shown the ability to adapt a model trained on one scene to new scenes on-the-fly [13]. Yet, [13] considers the problem of re-localization against a trajectory while we consider the problem of localization from a single image.

Learning absolute pose estimation. An alternative to regressing 2D-3D matches is to learn the complete localization pipeline, either via classification [82] or camera pose regression [30]–[32], [43], [47], [80]. These methods typically only require images and their corresponding camera poses as training data and minimize a loss on the predicted camera poses [31], [32]. However, using 2D-3D matches as part of the loss function can lead to more accurate results [31]. Similar to regressing 2D-3D matches, the learned representations are scene-specific and do not generalize. While methods that operate on individual images are not significantly more accurate than simple retrieval baselines [66], using image sequences for pose regression can significantly improve performance [54], [78]. In this paper, we however focus on the single-image case.

Learning relative pose estimation. In contrast to absolute pose estimation, which is a scene-specific task, learning to predict the relative pose between images is a problem that should generalize to unseen scenes. [77] propose a CNN that jointly predicts a depth map for one image and the relative pose w.r.t. a second image. In contrast to our approach, theirs requires depth maps for training. [87] is trained purely on a stream of images by using image synthesis as a supervisory loss function. The method is tested in an autonomous driving scenario that exhibits planar motion. Extending this method to the 6DOF scenario with larger baselines considered in this paper seems non-trivial.

[4] propose a network is jointly trained for the tasks of image retrieval (based on a novel frustum overlap distance) and relative camera pose regression. The latter is based on a $\mathbb{SE}(3)$ parameterization. Yet, the ability of [4] to generalize to new scenes is rather limited [66].

Most similar to our approach, [35] first identifies potentially relevant database images via image retrieval. A CNN is then used to regress the relative poses between the query and the retrieved images, followed by triangulation to estimate the query's absolute camera pose inside a RANSAC loop. [35] needs to find a weighting between the positional and rotational parts of the pose loss during training, which potentially needs to be adjusted per scene. We show that regressing essential matrices is a better choice. We also show how the resulting pose ambiguity can be handled via a novel RANSAC scheme. We analyze which parts of the localization pipeline fail when replaced by a data-driven approach, showing that learning the whole pipeline as in [35] is by far not the most accurate solution.

III. ESSENTIAL MATRIX BASED LOCALIZATION

In this section, we propose a scalable pipeline to estimate the absolute pose of a query image w.r.t. a scene represented by a database of images with known camera poses. Our pipeline, shown in Fig. 1, consists of three stages: (1) we use image retrieval to identify a set of images that potentially depict the same part of the scene as the query image (*c.f.* Sec. III-A). (2) for each retrieved image, we compute the essential matrix that encodes its relative pose w.r.t. the query image (*c.f.* Sec. III-B). (3) using the known absolute poses of the retrieved images and the essential matrices, we estimate the absolute pose of the query (*c.f.* Sec. III-C).

Why essential matrices? Since we are ultimately interested in extracting relative poses, one might wonder why not training a CNN to directly predict relative poses instead of essential matrices. Several works [35], [59] propose a model for relative pose prediction, with the main disadvantage of needing a scene-dependent hyperparameter (*c.f.* Sec. IV-B).

We notice that directly regressing essential matrix automatically resolves the scene-dependent weighting issue from relative pose regression and also leads to more accurate results. While directly decomposing the essential matrix into relative poses results in ambiguities, Sec. III-C shows how these ambiguities can be handled inside a RANSAC loop.

In the following, we describe our 3D model-free localization pipeline based on essential matrices, which is oblivious to the source of the essential matrices. Sec. IV then discusses multiple approaches to essential matrix estimation.

Notation. The absolute camera pose ($\mathtt{R}_{\mathscr{I}}$, $\mathbf{t}_{\mathscr{I}}$) of an image $\mathscr{I}$ is defined by a rotation matrix $\mathtt{R}_{\mathscr{I}}$ and a translation $\mathbf{t}_{\mathscr{I}}$ such that $\mathtt{R}_{\mathscr{I}}\mathbf{x}+\mathbf{t}_{\mathscr{I}}$ transforms a 3D point $\mathbf{x}$ from a global coordinate system into the local camera coordinate system of $\mathscr{I}$. Accordingly, the camera center $\mathbf{c}_{\mathscr{I}}$ of $\mathscr{I}$ in global coordinates is given by $\mathbf{c}_{\mathscr{I}} = -\mathtt{R}_{\mathscr{I}}^{T}\mathbf{t}_{\mathscr{I}}$. Notice that in practice,

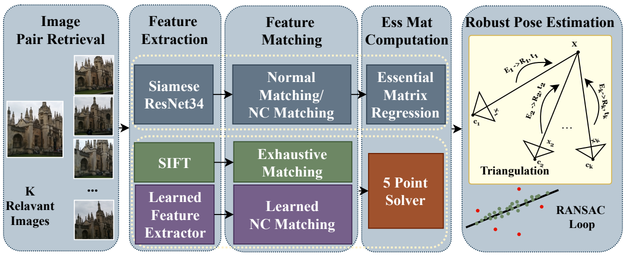


Fig. 1. Our localization pipeline: The pipeline first retrieves top-k similar training images for a query image using DenseVLAD descriptors, composing k input pairs. In the next stage, one of 3 approaches(Sec. IV) is used to estimate k essential matrices, which are fed into our RANSAC loop for relative pose extraction as well as the absolute pose computation.

we are representing the rotation $\mathtt{R}_{\mathscr{I}}$ by a quaternion $\mathbf{q}_{\mathscr{I}}$. As such, we will interchangeably use either a rotation matrix $\mathtt{R}$ or a quaternion $\mathbf{q}$ to denote a (relative or absolute) rotation.

A. Retrieving Relevant Database Images

We perform image retrieval by representing each database image by 4096-dimensional DenseVLAD descriptors [75], which has been shown to work under challenging conditions [64]. Compared to other learned pipelines for image retrieval [52], [53], DenseVLAD shows better generalization to unseen scenes, which fits well to our pipeline.

Pair selection. Simply picking top-k ranked retrieved images for each query image is not sufficient to obtain good performance. The top-k retrieved images are often taken from very similar poses, which causes problems when we estimate the camera position of the query via triangulation. We want to ensure larger triangulation angles while still keeping enough visual overlap for successful relative pose estimation. Starting with the top retrieved image, we thus iteratively select retrieved images to have certain minimal/maximal distances to the previously selected ones. The resulting query-database image pairs are then used for essential matrix estimation (*c.f.* Sec. IV-B). We handle outlier pairs, *i.e.*, database images depicting a different part of the scene than the query, robustly via RANSAC [21] as detailed in Sec. III-C.

B. Pairwise Relative Pose Estimation

For each image pair, we compute the essential matrix $\mathtt{E}$ that encodes the relative pose between the query and database image. Next, we extract the four relative poses $(\mathtt{R}, \mathbf{t})$, $(\mathtt{R}, -\mathbf{t})$, $(\mathtt{R}', \mathbf{t})$, $(\mathtt{R}', -\mathbf{t})$ corresponding to $\mathtt{E}$ [25], where $\mathtt{R}$ and $\mathtt{R}'$ are related by a 180° rotation around the baseline [25]. Traditionally, a *cheirality test* based on feature matches is used to find the correct relative pose among the four candidates [25]. However, methods that directly regress the relative pose typically do not provide such matches.

We use triangulation based on the estimated relative poses and the known absolute poses of the database images to estimate the position from which the query image was taken. We thus only need to disambiguate the two rotations, since the position of a point triangulated from multiple directions $\mathbf{t}_1, \ldots, \mathbf{t}_n$ does not change when flipping the signs of any direction $\mathbf{t}_i$. Thus, the absolute position of the query image can be uniquely determined by $n \geq 2$ images pairs.

Let $\mathtt{R}_i$ and $\mathtt{R}'_i$ be the possible relative rotations that transform from the local system of i-th retrieved image $\mathscr{I}_i$ to the local of the query image $\mathscr{I}_q$. Thus, the absolute rotation part of the query image is either $\mathtt{R}_i\mathtt{R}_{\mathscr{I}_i}$ or $\mathtt{R}'_i\mathtt{R}_{\mathscr{I}_i}$. Counting also the relative rotations estimated from another image pair $(\mathscr{I}_j, \mathscr{I}_q)$, we get four absolute rotation predictions $\mathtt{R}_i\mathtt{R}_{\mathscr{I}_i}$, $\mathtt{R}'_i\mathtt{R}_{\mathscr{I}_i}$, $\mathtt{R}_j\mathtt{R}_{\mathscr{I}_j}$, $\mathtt{R}'_j\mathtt{R}_{\mathscr{I}_j}$. In theory, two of them will be identical, while the others differ largely from each other, since $\mathtt{R}_i$ and $\mathtt{R}'_i$ (also $\mathtt{R}_j$ and $\mathtt{R}'_j$) are related by a 180° rotation. In practice, we consider the relative rotations (from each pair) that corresponds the two absolute pose predictions with smallest angle difference to be true ones.

C. Absolute Pose Estimation via RANSAC

Consider a pair $((\mathscr{I}_i, \mathscr{I}_q), (\mathscr{I}_j, \mathscr{I}_q))$ of image pairs. Let $\mathtt{R}_q$ be the absolute rotation of the query image estimated from the two image pairs as described above. Let $\mathtt{R}_i$ and $\mathtt{R}_j$ be the relative rotations consistent with $\mathtt{R}_q$. Using the two relative translation directions $\mathbf{t}_i$ and $\mathbf{t}_j$, we can determine the position of the query image via triangulation from the two rays $\mathbf{c}_{\mathscr{I}_i} + \lambda_i \mathtt{R}^T_{\mathscr{I}_i}\mathtt{R}_i\mathbf{t}_i$ and $\mathbf{c}_{\mathscr{I}_j} + \lambda_j \mathtt{R}^T_{\mathscr{I}_j}\mathtt{R}_j\mathbf{t}_j$, where λ_i, $\lambda_j \in \mathbb{R}$ define point positions along the rays. The result of triangulation is only defined if the three camera centers are not collinear. In practice, we use more than two database images to compute the final pose. Hence, this will only become a problem in scenarios where all images are taken exactly on a line, *e.g.*, a self-driving car is driving exactly the same route on a line.

As shown above, a hypothesis for the absolute pose of a query image can be estimated from two pairs. To be robust to outlier pairs, we use RANSAC [21]: In each iteration, we sample two pairs $(\mathscr{I}_i, \mathscr{I}_q)$, $(\mathscr{I}_j, \mathscr{I}_q)$ and use them to estimate the absolute pose hypothesis $(\mathtt{R}_{\mathscr{I}_q}, \mathbf{t}_{\mathscr{I}_q})$ of the query image. Next, we determine which image pairs are inliers to that pose hypothesis. For a pair $(\mathscr{I}_k, \mathscr{I}_q)$ defining four relative poses between $\mathscr{I}_k$ and $\mathscr{I}_q$, we first determine the relative rotation $\mathtt{R}_k$ that minimizes the angle between the absolute rotations $\mathtt{R}_{\mathscr{I}_q}$ and $\mathtt{R}_k\mathtt{R}_{\mathscr{I}_k}$. We then use $\mathtt{R}_k$ to measure how consistent the predicted relative pose is with the absolute pose hypothesis predicted by the image pair. The relative translation from $\mathscr{I}_q$ to $\mathscr{I}_k$ predicted by the pose $(\mathtt{R}_{\mathscr{I}_q}, \mathbf{t}_{\mathscr{I}_q})$ is given as $\mathbf{t}_{\text{pred}} = \mathtt{R}_{\mathscr{I}_k}(\mathbf{c}_{\mathscr{I}_q} - \mathbf{c}_{\mathscr{I}_k})$.

We measure the consistency with the predicted relative translation $\mathbf{t}_k$ via the angle $\alpha = \cos^{-1}(\mathbf{t}_k^T\mathbf{t}_{\text{pred}}/(||\mathbf{t}_k||_2||\mathbf{t}_{\text{pred}}||_2)$. If the angle between the two predicted translation directions is below a given threshold $\alpha_{\max}$, we consider the pair $(\mathscr{I}_k, \mathscr{I}_q)$ as an inlier.

We use local optimization [36] inside RANSAC to better account for noisy relative pose predictions. RANSAC finally returns the pose with the largest number of inliers.

IV. ESSENTIAL MATRIX ESTIMATION

While Deep Learning has made huge advances in other vision tasks such as image classification, in visual localiation end-to-end trained methods are still far behind classic methods in terms of accuracy [66]. We are highly interested in understanding how to better leverage the power of data-driven methods to build a robust, scalable, flexible and generalizeable localization pipeline. To this end, we propose different approaches for essential matrix estimation, ranging from purely hand-crafted to purely data-driven models.

A. Feature-based: SIFT + 5-Point Solver

Assuming known camera intrinsics, a classical approach uses local features (in our case SIFT [41]) to establish 2D-2D matches between a query and a database image. These matches are then used to estimate the essential matrix by applying the 5-point solver [48] inside a RANSAC loop. This approach, which does not need a 3D scene model, serves as a baseline within our localization pipeline (*c.f.* Sec. III).

B. Learning-based: Direct Regression via EssNet

The modern alternative to the classical pipeline is to train a CNN for relative pose regression. In the following, we introduce our approach based on essential matrices and discuss its advantages over existing methods.

Relative pose parametrization. Inspired by work on absolute pose regression, [4], [35], [44] propose to directly regress the relative poses with siamese neural networks. [35], [44] parametrize the pose via a rotation and a translation and use the following weighted loss during training

$$\mathscr{L}_w(y^*, y) = \|\mathbf{t} - \mathbf{t}^*\|_2 + \beta \, \|\mathbf{q} - \mathbf{q}^*\|_2 \quad . \tag{1}$$

Here, $y^* = (\mathbf{q}^*, \mathbf{t}^*)$ is the relative pose label, $y = (\mathbf{q}, \mathbf{t})$ is the relative pose prediction, $\mathbf{q}$ is the relative rotation encoded in a 4D quaternion, and $\mathbf{t}$ is the 3D translation. Notice, the β in $\mathscr{L}_w$ is a hyperparameter to balance the learning between translation and rotation, which is scene-dependent (*e.g.*, its values differ significantly for indoor and outdoor scenes [32]). [44] performs grid search to find the optimal β, following other absolute pose methods [30], [32], [80]. [35] note that setting $\beta = 1.0$ works well for indoor scenes. Yet, they do not provide any results for outdoor scenes, where finding a single suitable weighting factor is harder due to larger variations in the distance of the camera to the scene. [31] propose to learn the weighting parameter β from training data, but are also restricted to a single parameter.

The need for the hyperparameter β arises as the rotation (in degrees) and translation (in meters) are typically in different units. We note that it can be eliminated through regressing essential matrices, which implicitly define a weighting between an orthonormal rotation matrix and a unit-norm translation vector. Tab. II shows that our method based on essential matrix outperforms [35], verifying our approach.

Network architecture. We use a siamese neural network based on ResNet34 [27] (until the last average pooling layer) as our backbone (as in [8], [35]). While [35], [44] directly regress relative poses from the concatenated feature maps using with the weighted loss function defined in Eq. 1, we first involve a *feature matching* step that resembles the process in classic feature-based localization methods. We analyze two options for the matching step: 1) a simple fixed matching layer [56] (**EssNet**), essentially a matrix dot product between feature maps coming from the two images. 2) a learnable Neighbordhood Consensus (NC) matching layer [57] (**NC-EssNet**), which enforces local geometric consistency on the matches. Both matching versions combine the two feature maps produced by ResNet into a single feature tensor that can be seen as a matching score map.

The score map is fed to regression layers to predict the essential matrix. The regression layers consist of two blocks of convolutional layers followed by batch normalization with ReLU, and finally a fully connected layer to regress a 9D vector which approximates the essential matrix. This approximation is then projected to a valid essential matrix by replacing the first two singular values of the approximation with their mean value and finally sets the smallest singular value to 0. We use standard functionality provided by PyTorch [49] for SVD backpropagation.

Loss function. During training, we minimize the Euclidean distance between the predicted $\mathtt{E}$ and the ground truth essential matrix $\mathtt{E}^*$:

$$\mathscr{L}_{ess}(\mathtt{E}^*, \mathtt{E}) = \|\mathbf{e} - \mathbf{e}^*\|_2 \, . \tag{2}$$

Here, $\mathbf{e} \in \mathbb{R}^9$ is the vectorized $\mathtt{E} \in \mathbb{R}^{3\times3}$. Given a relative pose label $(\mathtt{R}^*, \mathbf{t}^*)$, the ground truth essential matrix is $\mathtt{E}^* = [\mathbf{t}^*]_\times \mathbb{R}^*$, where $[\mathbf{t}^*]_\times$ is the skew-symmetric matrix of the normalized translation label $\mathbf{t}^*$, *i.e.*, $||\mathbf{t}^*|| = 1$.

C. Hybrid: Learnable Matching + 5-Point Solver

As a combination of the classical and the regression approaches, we propose a hybrid method: Feature extraction and matching are learned via neural networks, resulting in a set of 2D-2D matches. The 5-point algorithm inside a RANSAC loop is then used to compute the essential matrix. In terms of architecture, this approach is equivalent to NC-EssNet without the regression layers.

V. EXPERIMENTS

In the following, we evaluate our novel localization approach based on essential matrix estimation. In particular, we are interested in using our approach to analyze why methods based on relative pose regression do not generalize as theoretically expected. To this end, we first demonstrate that our approach, based on handcrafted features and RANSAC-based essential matrix estimation, achieves state-of-the-art performance (*c.f.* Sec. V-A). We then use learned essential matrix estimation approaches inside our framework to analyze their weaknesses (*c.f.* Sec. V-B). Finally, Sec. V-C discusses our results and draws conclusions for future work.

Datasets and evaluation protocol. We follow common practice and use the Cambridge Landmarks [32] and 7 Scenes [70] datasets for evaluation. For both datasets and all methods, we report the median absolute position error in meters and the median absolute rotation error in degrees, averaged over all scenes within the dataset.

Implementation details. We split 1/6 of the training set images as validation images to control the training process. Training pairs are generated through image retrieval using the CNN (resnet101-gem) proposed in [53].

TABLE I

RANSAC thresholds used in our experiments.

Method/Scenes	(NC-)EssNet	SIFT+5Pt	Learnable Matching +5Pt
Indoor(t1/t2)	-/5	0.5/15	5.5/20
Outdoor(t1/t2)	-/5	0.5/5	4.0/15

TABLE II

Results on Cambridge Landmarks [32] and 7 Scenes [70]. We report the average median position [m] / orientation [°] errors. Methods marked with a * build a scene-specific representation and do not generalize to unseen scenes.

		Cambridge Landmarks	7 Scenes
IR	DenseVLAD [75]	2.56/7.12	0.26/13.11
	DenseVLAD + Inter. [66]	1.67/4.87	0.24/11.72
3D	*Active Search [63]	0.29/0.63	0.05/2.46
	*DSAC++ [6]	**0.14/0.33**	**0.04/1.10**
APR	*PoseNet (PN) [32]	2.09/6.84	0.44/10.44
	*Learn. PN [31]	1.43/2.85	0.24/7.87
	*Bay. PN [30]	1.92/6.28	0.47/9.81
	*Geo. PN [31]	1.63/2.86	0.23/8.12
	*LSTM PN [80]	1.30/5.52	0.31/9.85
	*MapNet [8]	1.63/3.64	0.21/7.78
	*MapNet+PGO [8]	-	0.18/6.56
RPR	Relative PN [35] (U)	-	0.36/18.37
	Relative PN [35] (7S)	-	0.21/9.28
	RelocNet [4] (SN)	-	0.29/11.29
	RelocNet [4] (7S)	-	0.21/6.73
	*AnchorNet [59]	0.84/2.10	0.09/6.74
Ours	Sift+5Pt	0.47/0.88	0.08/1.99
	EssNet	1.08/3.41	0.22/8.03
	NC-EssNet	0.85/2.82	0.21/7.50
	NC-EssNet(7S)+NCM+5Pt	0.89/1.39	0.19/4.28
	Imagenet+NCM+5Pt	0.83/1.36	0.19/4.30
	EssNet224(SN)+NCM+5Pt	0.90/1.37	0.19/4.35

EssNet and NC-EssNet are trained with exact the same settings for fair comparison. We use a ResNet34 pre-trained on ImageNet [17]. The regression network layers are initialized with Kaiming initialization [26]. For each dataset, we train the model on training pairs from *all scenes* and evaluate per scene at test time. Note, that we use a single network to test on all Cambridge Landmarks sequences, while absolute pose methods [31], [32], [47], [80] train a separate network per sequence. All training images are first rescaled so that the shorter side has 480 pixels and then random cropped for training and center cropped for testing to 448×448 pixels. All models are trained using the AdamOptimizer [33] with learning rate $1e^{-4}$ and weight decay $1e^{-6}$ in a batch size of 16 for at most 200 epochs. We early stop training if overfitting is observed and use the model with best validation accuracy. The code is implemented using Pytorch [49] and executed on NVidia TITAN Xp GPUs.

During testing, we use DenseVLAD [75] to identify the top-5 most similar training images for each query. The retrieved images have to satisfy the following condition designed to avoid retrieving close-by views and thus acute triangulation angles: Starting from the top-ranked image, we select the next image that has a distance within $[a,b]$ meters to all previously selected images. For outdoor scenes $a = 3, b = 50$ and for indoor scenes $a = 0.05, b = 10$. We show the choice of RANSAC thresholds $t1$ in the 5-point algorithm [48] to distinguish inliers and outliers and $t2$ in our RANSAC algorithm (*c.f.* Sec. III-C) to remove outlier pairs for absolute pose estimation in Tab. I. The thresholds were chosen through grid-search.

A. Comparison with State-of-the-Art

To validate our pipeline based on essential matrices, we compare results obtained when using SIFT features and the 5-point solver for estimating the essential matrices (*c.f.* Sec. IV-A) to state-of-the-art methods. We use COLMAP [67] to extract and match features and the 5-point RANSAC implementation provided in OpenCV [7]. We compare our approach to methods for absolute pose regression (APR) [8], [30]–[32], [80], relative pose regression (RPR) [4], [35], [59], the two image retrieval (IR) baselines based on DenseVLAD [75][1] used in [66], and two state-of-the-art structure-based methods (3D) that explicitly estimate 2D-3D matches [6], [63]. For two RPR methods, we report results obtained when training on the 7 Scenes dataset (7S) and when training on an unrelated dataset (University (U) [35] or ScanNet (SN) [15]).

Tab. II shows that our approach (SIFT+5Pt) consistently outperforms all IR, APR and RPR methods, validating the effectiveness of our pipeline. Compared to structure-based methods (3D), our approach performs competitively when taking into account that both Active Search and DSAC++ need to build a scene-specific model. In contrast, our approach just operates on posed images without the need for using any 3D structure. Note that DSAC++ requires two or more days of training while our approach is light-weight and does not require any training.

B. Analyzing Relative Pose Regression (RPR)

One motivation for our localization pipeline is to understand why RPR methods perform worse compared to structure-based methods. In the following experiment, we use (NC-)EssNet (*c.f.* Sec. IV-B) as the RPR method inside our pipeline.

Comparison with state-of-the-art. Tab. II compares our approaches against the current state-of-the-art. For visibility, we mark results that are less accurate than NC-EssNet in red. As can be seen, NC-EssNet, our best regression model, performs better than all APR approaches except for MapNet+PGO which uses external GPS information. Also, our NC-EssNet is competitive to RelocNet and outperforms Relative PN. While NC-EssNet is less accurate than AnchorNet [59], AnchorNet needs to be trained explicitly per scene as it encodes training images in the network. The results show that our methods achieve state-of-the-art performance among pose regression methods.

Failure to generalize. Compared to absolute pose regression, the promise of relative pose regression is generalization to new scenes [4], [35]: An absolute pose estimate is scene-specific as it depends on the coordinate system used. In contrast, a network that learns to regress a pose relative to another image could learn general principles that are

[1] DenseVLAD + Inter. denotes interpolating between the top-ranked database images. See [66] for details.

TABLE III

GENERALIZATION STUDY OF REGRESSION MODELS. WE SHOW AVERAGE MEDIAN POSITION (IN METERS) / ORIENTATION (IN DEGREES) ERRORS.

Essential Matrix Estimation	Training Data	Testing Data Cambridge	7Scenes
EssNet	Cambridge	1.08/3.41	0.57/80.06
NC-EssNet	Cambridge	0.85/2.82	0.48/32.97
EssNet	7Scenes	10.36/85.75	0.22/8.03
NC-EssNet	7Scenes	7.98/24.35	0.21/7.50
SIFT+5Pt	-	0.47/0.88	0.08/1.99

TABLE IV

EVALUATING THE IMPACT OF TRAINED FEATURES ON LOCALIZATION PERFORMANCE WHEN USING DIFFERENT TRAINING STRATEGIES.

Feature Matching	Train Task	Train Data	Cambridge	7Scenes
ImageNet+NCM	IC	ImageNet [58]	0.83/1.36	0.19/4.30
NC-EssNet+NCM	EMR	7S	0.89/1.39	0.19/4.28
NC-EssNet+NCM	EMR	CL	0.96/1.43	0.20/4.61
EssNet224+NCM	EMR	MD [39]	0.98/1.4	0.20/4.70
EssNet224+NCM	EMR	SN [15]	0.90/1.37	0.19/4.35
EssNet224+NCM	EMR	MD+7S+CL	0.96/1.48	0.23/4.89
SIFT+5Pt	-	-	0.47/0.88	0.08/1.99

applicable to unseen scenes.

Tab. III analyzes the ability of EssNet and NC-EssNet to generalize from indoor to outdoor scenes and vice versa, where we mark failure cases in purple. As can be seen, there is a substantial gap in pose accuracy compared to training on the same scenes and especially compared to the classical variant (SIFT+5Pt) of our pipeline. This clearly indicates that EssNet and NC-EssNet fail to learn a general underlying principle. As similar observation holds for [4], [35] in Tab. II, based on the performance when trained on 7 Scenes (7S) and on another dataset (U or SN).

Looking at Tab. III and Tab. II, the important question to ask is why RPR methods fail to generalize: Do the features extracted in their base networks fail to generalize, is there a lack of generalization in the layer that regresses the relative pose, or is it a combination of both? In order to better understand the behavior of EssNet and NC-EssNet, we consider the hybrid version of our pipeline (*c.f.* Sec. IV-C).

The hybrid variant *always* uses the NC matching layer (NCM) trained on the unrelated ivd dataset [57] to extract feature matches. To analyze the impact of the **feature extraction** on the generalization performance, we compare our *ResNet34* backbones trained in different ways and on multiple datasets, *e.g.*, the pretrained model for the image classification (IC) task on ImageNet [58] and our trained models for the essential matrix regression (EMR) task on MegaDepth(MD) [39] (outdoor), 7 Scenes(7S) [70] (indoor), and Cambridge Landmarks(CL) [32] (outdoor) datasets. In order to make training computationally feasible on large datasets such as MegaDepth and ScanNet, we train EssNet with reduced image resolution (224×224). We denote these trained *feature extractors* with EssNet224. Note that for our hybrid, we perform inference with the original high resolution images.

Tab. IV evaluates the performance of the different training strategies on the localization accuracy of our hybrid approach. As can be seen, there is little variation in performance independently how the features are trained. This clearly shows that the features themselves generalize well and that the failure to generalize observed in Tab. III is caused by the regression layers.

C. Discussion

In a classical approach [25], the relative pose between two images is estimated by finding feature correspondences in the image pair. When directly regressing the relative pose/essential matrix from an image pair, we can only assume that an implicit feature matching is performed within regression. In contrast, our hybrid approach explicitly learns the feature matching task and adopts the established multi-view geometry knowledge to compute relative poses from correspondences. The fact that the relative pose regression layers fail to generalize to unseen scenes and to produce accurate poses implies that the implicit matching cannot be properly learned by a regression network. While explicitly learning the matching task leads to better generalization, the resulting poses are still not as accurate as the poses estimated by SIFT+5pt, as can be seen in Tab. IV. Such inaccuracy is related to the fact that the current CNN features are coarsely localized on the images, that is, the features from later layers are not mapped to a single pixel but rather an image patch. One possible solution would be networks designed to obtain better localized features [18], [20]. Another would be to follow [16], [46], [55], where a network is trained to detect outliers, and can be applied as a post-processing step for any type of matches. However, integrating those methods into an end-to-end pipeline going from image pairs to poses is not straight-forward and will constitute interesting future work.

VI. CONCLUSION

In this paper, we have proposed a novel framework for visual localization from essential matrices. Our approach is light-weight and flexible in the sense that it does not use information about the 3D scene structure model of the scene and can thus easily be applied to new scenes. Our results show that our framework can achieve state-of-the-art results. We have evaluated our framework using three different methods for computing essential matrices, ranging from purely hand-crafted to purely data-driven. By comparing their results, we have shown that the purely data-driven approach does not generalize well and have identified the reason for this failure as the relative pose regression layers. Furthermore, we have shown that the features and matches used by the data-driven approach themselves generalize quite well. However, directly using them for pose estimation yields less accurate results compared to the hand-crafted version of our pipeline. Based on our analysis, it is clear that more research is required before data-driven visual localization methods perform accurately and easily generalize to new scenes.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.

[2] R. Arandjelović and A. Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition . In *ACCV*, 2014.

[3] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide area localization on mobile phones. In *ISMAR*, 2009.

[4] V. Balntas, S. Li, and V. Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[5] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC - Differentiable RANSAC for Camera Localization. In *CVPR*, 2017.

[6] E. Brachmann and C. Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018.

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] F. Camposeco, A. Cohen, M. Pollefeys, and T. Sattler. Hybrid Scene Compression for Visual Localization. In *CVPR*, 2018.

[10] S. Cao and N. Snavely. Graph-Based Discriminative Learning for Location Recognition. In *CVPR*, 2013.

[11] S. Cao and N. Snavely. Minimal Scene Descriptions from Structure from Motion Models. In *CVPR*, 2014.

[12] R. O. Castle, G. Klein, and D. W. Murray. Video-rate Localization in Multiple Maps for Wearable Augmented Reality. In *ISWC*, 2008.

[13] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, L. Di Stefano, and P. H. S. Torr. On-The-Fly Adaptation of Regression Forests for Online Camera Relocalisation. In *CVPR*, 2017.

[14] S. Choudhary and P. J. Narayanan. Visibility probability structure from sfm datasets and applications. In *ECCV*, 2012.

[15] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[16] Z. Dang, K. Moo Yi, Y. Hu, F. Wang, P. Fua, and M. Salzmann. Eigendecomposition-free training of deep networks with zero eigenvalue-based losses. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[18] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[19] M. Donoser and D. Schmalstieg. Discriminative Feature-to-Point Matching in Image-Based Locallization. In *CVPR*, 2014.

[20] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[21] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981.

[22] M. Geppert, P. Liu, Z. Cui, M. Pollefeys, and T. Sattler. Efficient 2d-3d matching for multi-camera visual localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5972–5978. IEEE, 2019.

[23] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla. Learning per-location classifiers for visual place recognition. In *CVPR*, 2013.

[24] A. Guzman-Rivera, P. Kohli, B. Glocker, J. Shotton, T. Sharp, A. Fitzgibbon, and S. Izadi. Multi-Output Learning for Camera Relocalization. In *CVPR*, 2014.

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

[26] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[28] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *CVPR*, 2009.

[29] H. Jegou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *ECCV*, 2008.

[30] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016.

[31] A. Kendall and R. Cipolla. Geometric Loss Functions for Camera Pose Regression With Deep Learning. In *CVPR*, 2017.

[32] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015.

[33] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[34] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR*, 2011.

[35] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala. Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network. In *ICCV Workshops*, 2017.

[36] K. Lebeda, J. Matas, and O. Chum. Fixing the Locally Optimized RANSAC. In *BMVC*, 2012.

[37] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*, 2012.

[38] Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition Using Prioritized Feature Matching. In *ECCV*, 2010.

[39] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[40] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *CVPR*, 2012.

[41] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[42] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart. Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. In *RSS*, 2015.

[43] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based Localization using Hourglass Networks. In *ICCV Workshops*, 2017.

[44] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 675–687, 2017.

[45] L. Meng, J. Chen, F. Tung, J. J. Little, J. Valentin, and C. W. de Silva. Backtracking Regression Forests for Accurate Camera Relocalization. In *IROS*, 2017.

[46] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua. Learning to Find Good Correspondences. In *CVPR*, 2018.

[47] T. Naseer and W. Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *IROS*, 2017.

[48] D. Nister. An efficient solution to the five-point relative pose problem. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–195. IEEE, 2003.

[49] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[50] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[51] N. Piasco, D. Sidibé, V. Gouet-Brunet, and C. Demonceaux. Learning scene geometry for visual localization in challenging conditions. In *ICRA*, 2019.

[52] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, pages 3–20, 2016.

[53] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *TPAMI*, 2018.

[54] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics And Automation Letters (RA-L)*, 3(4):4407–4414, 2018.

[55] R. Ranftl and V. Koltun. Deep fundamental matrix estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018.

[56] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[57] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic.

Neighbourhood consensus networks. In *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 2018.

[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[59] S. Saha, G. Varma, and C. V. Jawahar. Improved Visual Relocalization by Discovering Anchor Points. In *BMVC*, 2018.

[60] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[61] P.-E. Sarlin, F. Debraine, M. Dymczyk, and R. Siegwart. Leveraging deep visual descriptors for hierarchical efficient localization. In *Conference on Robot Learning*, pages 456–465, 2018.

[62] T. Sattler, M. Havlena, F. Radenović, K. Schindler, and M. Pollefeys. Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition. In *ICCV*, 2015.

[63] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *PAMI*, 39(9):1744–1756, 2017.

[64] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *CVPR*, 2018.

[65] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *CVPR*, 2017.

[66] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In *CVPR*, 2019.

[67] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[68] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic Visual Localization. In *CVPR*, 2018.

[69] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic Visual Localization. In *CVPR*, 2018.

[70] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. *CVPR*, 2013.

[71] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[72] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *PAMI*, 39(7):1455–1461, 2017.

[73] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *CVPR*, 2018.

[74] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl. Semantic Match Consistency for Long-Term Visual Localization. In *ECCV*, 2018.

[75] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015.

[76] A. Torii, J. Sivic, and T. Pajdla. Visual localization by linear combination of image descriptors. In *ICCVW*, 2011.

[77] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. DeMoN: Depth and Motion Network for Learning Monocular Stereo. In *CVPR*, 2017.

[78] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *ICRA*, May 2018.

[79] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. Torr. Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization. In *CVPR*, 2015.

[80] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *ICCV*, 2017.

[81] P. Weinzaepfel, G. Csurka, Y. Cabon, and M. Humenberger. Visual localization by learning objects-of-interest dense match regression. In *CVPR*, 2019.

[82] T. Weyand, I. Kostrikov, and J. Phiblin. Planet - photo geolocation with convolutional neural networks. *ECCV*, 2016.

[83] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *ECCV*, 2010.

[84] A. R. Zamir and M. Shah. Image Geo-Localization Based on Multiple Nearest Neighbor Feature Matching Using Generalized Graphs. *PAMI*, 36(8):1546–1558, 2014.

[85] B. Zeisl, T. Sattler, and M. Pollefeys. Camera pose voting for large-scale image-based localization. In *ICCV*, 2015.

[86] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, 2006.

[87] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised Learning of Depth and Ego-Motion From Video. In *CVPR*, 2017.

6 Patch2Pix: Epipolar-Guided Pixel-Level Correspondences

6.1 Summary

Finding accurate image correspondences is a crucial step in various computer vision applications, including Structure-from-Motion (SfM), Simultaneous Localization and Mapping (SLAM), and Visual Localization.

The conventional approach to visual localization involves three key stages: (i) local feature detection and description, (ii) feature matching, and (iii) outlier rejection. However, emerging correspondence networks attempt to integrate these steps into a single network, albeit with reduced matching resolution due to memory constraints.

To overcome the memory limitations of dense matching correspondence networks, we propose a novel perspective: a detect-to-refine strategy. This entails generating patch-level match proposals initially, followed by a refinement process. Our contribution, Patch2Pix, is a refinement network designed to enhance match proposals by regressing pixel-level correspondences within the local regions defined by these proposals. Additionally, it incorporates a joint outlier rejection mechanism based on confidence scores.

Patch2Pix is trained in a weakly supervised manner to ensure that the learned correspondences align with the epipolar geometry of the input image pair. Experimental results demonstrate a significant enhancement in the performance of correspondence networks across tasks such as image matching, homography estimation, and localization.

Moreover, our findings indicate that the knowledge gained from the learned refinement process can be applied to fully-supervised methods without requiring re-training. This breakthrough leads to state-of-the-art localization performance.

6.2 Author Contributions

The author of this dissertation significantly contributed to

- developing the main concepts
- implementing the algorithm
- evaluating the numerical experiments
- writing the paper

6.3 Preprint

Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2021, pp. 4669–4678

Following the IEEE reuse permissions, we include the *accepted* version of the publication. The published version available under https://doi.org/10.1109/CVPR46437.2021.00464.

Patch2Pix: Epipolar-Guided Pixel-Level Correspondences

Qunjie Zhou[1] Torsten Sattler[2] Laura Leal-Taixé[1]
[1]Technical University of Munich [2]CIIRC, Czech Technical University in Prague *

Abstract

The classical matching pipeline used for visual localization typically involves three steps: (i) local feature detection and description, (ii) feature matching, and (iii) outlier rejection. Recently emerged correspondence networks propose to perform those steps inside a single network but suffer from low matching resolution due to the memory bottleneck. In this work, we propose a new perspective to estimate correspondences in a detect-to-refine manner, where we first predict patch-level match proposals and then refine them. We present Patch2Pix, a novel refinement network that refines match proposals by regressing pixel-level matches from the local regions defined by those proposals and jointly rejecting outlier matches with confidence scores. Patch2Pix is weakly supervised to learn correspondences that are consistent with the epipolar geometry of an input image pair. We show that our refinement network significantly improves the performance of correspondence networks on image matching, homography estimation, and localization tasks. In addition, we show that our learned refinement generalizes to fully-supervised methods without re-training, which leads us to state-of-the-art localization performance. The code is available at `https://github.com/GrumpyZhou/patch2pix`.

1. Introduction

Finding image correspondences is a fundamental step in several computer vision tasks such as Structure-from-Motion (SfM) [36, 41] and Simultaneous Localization and Mapping (SLAM) [8, 24]. Given a pair of images, pixel-level correspondences are commonly established through a local feature matching pipeline, which involves the following three steps: i) detecting and describing local features, ii) matching the nearest neighbors using the feature descriptors, and iii) rejecting outlier matches.

Traditional hand-crafted local features such as SIFT [15] or SURF [2] are vulnerable to extreme illumination changes, motion blur and repetitive and weakly textured scenes. Therefore, recent works [5–7, 16, 17, 28, 40] propose to learn to detect and describe local features using neural networks, showing that learned features can be robustly matched under challenging conditions [6, 17, 28, 40]. Instead of focusing on improving local features, [3, 22, 38, 42] suggest to learn a filtering function from sets of correspondences to reject outlier matches. A recent method [33] further proposes to jointly learn the matching function and outlier rejection via graph neural networks and the Sinkhorn algorithm [4, 37]. Combining a learned feature [5] and learned matcher [33] has set the state-of-the-art results on several geometry tasks, showing a promising direction towards a full learnable matching pipeline.

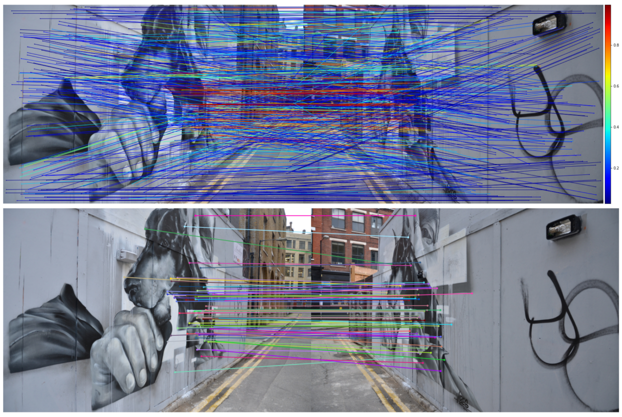


Figure 1. **An example of** ***Patch2Pix*** **correspondences.** In the top figure, the matches refined by *Patch2Pix* are coloured according to the predicted confidence scores. The less confident matches (in blue) appear mostly on the road or the blank wall. In the bottom figure, we show that the inlier matches can well handle the large viewpoint change. We show more quantitative results for handling various challenging conditions in the supp. mat (*c.f*. Sec. D).

Learning the whole matching pipeline has already been investigated in several works [13, 30, 31], where a single network directly outputs correspondences from an input image pair. The main challenge faced with those correspondence networks is how to efficiently perform matching while reaching pixel-level accuracy. In order to keep computation speed and memory footprint manageable, [29] has

*This research was funded by the Humboldt Foundation through the Sofja Kovalevskaya Award, the EU Horizon 2020 project RICAIP (grant agreeement No. 857306), and the European Regional Development Fund under project IMPACT (No. CZ.02.1.01/0.0/0.0/15 003/0000468).

to match at a rather low resolution, which is shown to be less accurate in relative pose estimation [43]. While sparse convolutions have been applied in [30] to match at higher resolution, they still do not achieve pixel-level matching. One advantage of the correspondences networks [30, 31] is that they are weakly supervised to maximize the average matching score for a matching pair and minimize it for a non-matching pair, however, they learn less effectively in pixel-level matching. This is in contrast to methods that require full supervision from ground truth (GT) correspondences [5, 6, 10, 17, 28, 33]. While the GT correspondences provide very precise signals for training, they might also add bias to the learning process. For example, using the sparse keypoints generated by an SfM pipeline with a specific detector as supervision, a keypoint detector might simply learn to replicate these detections rather than learning more general features [26]. To avoid such type of bias in the supervision, a recent work [40] proposes to use relative camera poses as weak supervision to learn local feature descriptors. Compared to the mean matching score loss used in [30, 31], they are more precise by containing the geometrical relations between the images pairs.

In this paper, we propose *Patch2Pix*, a new view for the design of correspondence networks. Inspired by the successful *detect-to-refine* practice in the object detection community [27], our network first obtains patch-level match proposals and then refines them to pixel-level matches. See an example of our matches in Fig. 1. Our novel refinement network is weakly supervised by epipolar geometry computed from relative camera poses, which are used to regress geometrically consistent pixel-wise matches within the patch proposal. Compared to [40], we optimize directly on match locations to learn matching, while they optimize through matching scores to learn feature descriptors. Our method is extensively evaluated on a set of geometry tasks, showing state-of-the-art results. We summarize our **contributions** as: i) We present a novel view for finding correspondences, where we first obtain patch-level match proposals and then refine them to pixel-level matches. ii) We develop a novel match refinement network that jointly refines the matches via regression and rejects outlier proposals. It is trained without the need for pixel-wise GT correspondences. iii) We show that our model consistently improves match accuracy of correspondence networks for image matching, homography estimation and visual localization. iv) Our model generalizes to fully supervised methods without the need for retraining, and achieves state-of-the-art results on indoor and outdoor long-term localization.

2. Related Work

Researchers have recently opted for leveraging deep learning to detect robust and discriminative local features [5–7, 17, 28, 40]. D2Net [6] detects keypoints by finding local maxima on CNN features at a 4-times lower resolution w.r.t. the input images, resulting in less accurate detections. Based on D2Net, ASLFeat [17] uses deformable convolutional networks and extracts feature maps at multiple levels to obtain pixel-level matches. R2D2 [28] uses dilated convolutions to preserve image resolution and predicts per-pixel keypoints and descriptors, which gains accuracy at the cost of computation and memory usage. Given the keypoints, CAPS [40] fuses features at several resolutions and obtains per-pixel descriptors by interpolation. The above methods are designed to learn local features and require a further matching step to predict the correspondences.

Matching and Outlier Rejection. Once local features are detected and described, correspondences can be obtained using Nearest Neighbor (NN) search [23] based on the Euclidean distance between the two feature representations. Outliers are normally filtered based on mutual consistency or matching scores. From a set of correspondences obtained by NN search, recent works [3, 22, 38, 42] learn networks to predict binary labels to identify outliers [22, 38, 42], or probabilities that can be used by RANSAC [9] to weight the input matches [3]. Notice, those methods do not learn the local features for matching and the matching function itself, thus they can only improve within the given set of correspondences. Recent works further propose to learn the whole matching function [10, 33]. SuperGlue [33] learns to improve SuperPoint [5] descriptors for matching using a graph neural network with attention and computes the correspondences using the Sinkhorn algorithm [4, 37]. S2DNet [10] extracts sparse features at SuperPoint keypoint locations for one image and matches them exhaustively to the dense features extracted for the other image to compute correspondences based on the peakness of similarity scores. While those methods optimize feature descriptors at keypoint locations specifically for the matching process, they do not solve the keypoint detection problem.

End-to-End Matching. Instead of solving feature detection, feature matching, and outlier rejection separately, recently correspondences networks [13, 30, 31] have emerged to accomplish all steps inside a single forward pass. NCNet uses a correlation layer [29] to perform the matching operation inside a network and further improves the matching scores by leveraging a neighborhood consistency score, which is obtained by a 4D convolution layer. Limited by the available memory, NCNet computes the correlation scores on feature maps with 16-times downscaled resolution, which has been proven not accurate enough for camera pose estimation [43]. SparseNCNet [30] uses a sparse representation of the correlation tensor by storing the top-10 similarity scores and replace dense 4D convolution with sparse convolutions. This allows SparseNCNet to obtain matches at 4-times downscaled resolution w.r.t. the original image. DualRC-Net [13], developed concurrently with our approach, outperforms SparseNCNet by combining the matching scores obtained from coarse-resolution and fine-resolution feature maps. Instead of refining the matching scores as in [13, 30], we use regression layers to refine the match locations at image resolution.

Full versus Weak Supervision. We consider methods that require information about exact correspondences to compute their loss function as fully supervised and those that do not need GT correspondences as weakly supervised. Most local feature detectors and descriptors are trained on exact correspondences either calculated using camera poses and depth maps [6, 10, 17] or using synthetic homography transformations [5, 28], except for CAPS [40] using epipolar geometry as weak supervision. Both S2DNet [10] and SuperGlue [33] requires GT correspondences to learn feature description and matching. Outlier filtering methods [3,22,38,42] are normally weakly supervised by the geometry transformations between the pair. DualRC-Net [13] is also fully supervised on exact correspondences, while the other two correspondence networks [30, 31] are weakly-supervised to optimize the mean matching score on the level of image pairs instead of individual matches. We use epipolar geometry as weak supervision to learn geometrically consistent correspondences where the coordinates of matches are directly regressed and optimize. In contrast, CAPS [40] uses the same level of supervision to learn feature descriptors and their loss optimizes through the matching scores whose indices give the match locations. We propose our two-stage matching network, based on the concept of learned correspondences [30, 31], which learns to predict geometrically consistent matches at image resolution.

3. Patch2Pix: Match Refinement Network

A benefit of correspondence networks is the potential to optimize the network directly for the feature matching objective without the need for explicitly defining keypoints. The feature detection and description are implicitly performed by the network and reflected in the found correspondences. However, there are two main issues causing the inaccuracy of the existing correspondence networks [30, 31]: i) the use of downscaled feature maps due to the memory bottleneck constrained by the size of the correlation map. This leads to every match being uncertain within two local patches. ii) Both NCNet [31] and SparseNCNet [30] have been trained with a weakly supervised loss which simply gives low scores for all matches of a non-matching pair and high scores for matches of a matching pair. This does not help identify good or bad matches, making the method unsuitable to locate pixel-accurate correspondences.

In order to fix those two sources of inaccuracies, we propose to perform matching in a two-stage *detect-to-refine* manner, which is inspired by two-step object detectors such as Faster R-CNN [27]. In the first correspondence detection stage, we adopt a correspondence network, *e.g.*, NC-Net, to predict a set of patch-level match proposals. As in Faster R-CNN, our second stage refines a match proposal in two ways: (i) using classification to identify whether a proposal is confident or not, and (ii) using regression to detect a match at pixel resolution within the local patches centered by the proposed match. Our intuition is that the correspondence network uses the high-level features to predict semantic matches at a patch-level, while our refinement network can focus on the details of the local structure to define more accurate locations for the correspondences. Finally, our network is trained with our weakly-supervised epipolar loss which enforces our matches to fulfill this geometric constraint defined by the relative camera pose. We name our network *Patch2Pix* since it predicts pixel-level matches from local patches, and the overview of the network architecture is depicted in Fig. 2. In the following, we take NC-Net as our baseline to obtain match proposals, yet we are not limited to correspondence networks to perform the match detection. We show later in our experiments that our refinement network also generalizes to other types of matching methods (*c.f.* Sec. 5.3 & 5.4). The following sections detail its architecture and training losses.

3.1. Refinement: Pixel-level Matching

Feature Extraction. Given a pair of images (I_A, I_B), a CNN backbone with L layers extracts the feature maps from each image. We consider $\{f_1^A\}_{l=0}^L$ and $\{f_l^B\}_{l=0}^L$ to be the activation maps at layer l for images I_A and I_B, respectively. At the layer index $l = 0$, the feature map is the input image itself, *i.e.*, $f_0^A = I_A$ and $f_0^B = I_B$. For an image with spatial resolution $H \times W$, the spatial dimension of feature map f_l is $H/2^l \times W/2^l$ for $l \in [0, L-1]$. For the last layer, we set the convolution stride as 1 to prevent losing too much resolution. The feature maps are extracted once and used in both the correspondence detection and refinement stages. The detection stage uses only the last layer features which contain more high-level information, while the refinement stage uses the features before the last layer, which contain more low-level details.

From match proposals to patches. Given a match proposal $m_i = (p_i^A, p_i^B) = (x_i^A, y_i^A, x_i^B, y_i^B)$, the goal of our refinement stage is to find accurate matches on the pixel level by searching for a pixel-wise match inside local regions. As the proposals were matched on a downscaled feature map, an error by one pixel in the feature map leads to inaccuracy of 2^{L-1} pixels in the images. Therefore, we define the search region as the $S \times S$ local patches centered at p_i^A and p_i^B, where we consider $S > 2^{L-1}$ to cover a larger region than the original $2^{L-1} \times 2^{L-1}$ local patches. Once

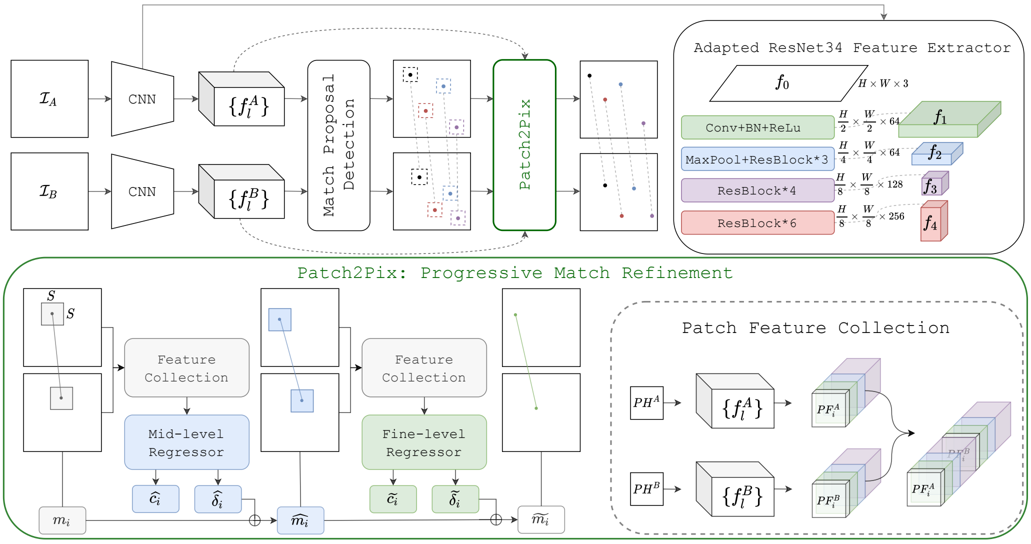


Figure 2. **Correspondence Refinement with** ***Patch2Pix*****.** *Top:* For a pair of images, features are first extracted using our adapted ResNet34 backbone and fed into a correspondence network, *e.g.*, NC matching layer [31], to detect match proposals. Those proposals are then refined by *Patch2Pix*, which re-uses the extracted feature maps. *Bottom:* We design two levels of regressors with the same architecture to progressively refine the match proposals at image resolution. For a pair of $S \times S$ local patches centered at a match proposal m_i, the features of the patches are collected as the input to our mid-level regressor to output (i) a confidence score $\widehat{c}_i$ which indicates the quality of the match proposal and (ii) a pixel-level local match $\widehat{\delta}_i$ found within the local patches. The updated match proposal $\widehat{m_i}$ updates the search space accordingly through a new pair of local patches. The fine-level regressor outputs the final confidence score $\widetilde{c}_i$ and $\widetilde{\delta}_i$ to obtained the final pixel-accurate match $\widetilde{m_i}$. The whole network is trained under weak supervision without the need for explicit GT correspondences.

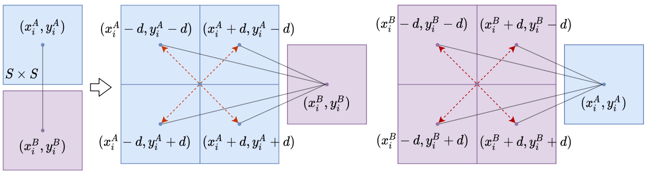


Figure 3. **Patch Expansion.** Given a match proposal $p_i^A = (x_i^A, y_i^A)$ and $p_i^B = (x_i^B, y_i^B)$, we move p_i^A towards its four corners by moving along the x- and y-axes by d pixels, which are matched to p_i^B to compose 4 new match proposals. Repeating it also from p_i^B to p_i^A, leads to 8 match proposals in total, which allows us to search in two $2S \times 2S$ local regions, compared to the original $S \times S$ patches.

we obtain a set of local patch pairs for all match proposals, the pixel-level matches are regressed by our network from the feature maps of the local patch pairs. We describe each component in detail below.

Local Patch Expansion. We further propose a patch expansion mechanism to expand the search region by including the neighboring regions, as illustrated in Fig. 3. We first move p_i^A towards its four corners along the x- and y-axes, each by d pixels. This gives us four anchor points for p_i^A that we match to p_i^B to compose four new match proposals. Similarly, we also expand p_i^B to get its four corner anchors and match them to p_i^A, giving us another four new match proposals. In the end, the expanded eight proposals identify eight pairs of $S \times S$ local patches. We set $d = S/2$ pixels so that the expanded search region defined by the expanded patches has size $2S \times 2S$ and still covers the original $S \times S$ searching space. The patch expansion to the patch proposals M_{patch} is especially useful during training since the network is forced to identify the correct proposal among spatially close and similar features.We show in the supp. mat (Sec. B) that our expansion mechanism can speed up the learning process and also improves the model performance. While one can also apply it during the inference to increase the search region, it will lead to a higher computation overhead. We thus refrain from using it during testing.

Progressive Match Regression. In order to locate pixel-level matches, we define the refinement task as finding a good match inside the pair of local patches. We achieve this using two regressors with the same architecture, *i.e.*, the mid-level and the fine-level regressor, to progressively identify the final match, which is shown in the lower part of Fig. 2. Given a pair of $S \times S$ patches, we first collect the corresponding feature information from previously extracted activation maps, *i.e.*, $\{f_l^A\}, \{f_l^B\}$. For every point location (x, y) on the patch, its corresponding location on the l-layer feature map is $(x/2^l, y/2^l)$. We select all features from the layers $\{0, \dots, L-1\}$ and concatenate them into a single feature vector. The two gathered feature patches PF_i^A and PF_i^B are concatenated along the feature dimension and fed into our mid-level regressor. The regressor first aggregates the input features with two convolutional layers into a compact feature vector, which is then processed by two fully connected (fc) layers, and finally outputs our network predictions from two heads implemented as two fc layers. The first head is a regression head, which outputs a set of local matches $\widehat{M_\Delta} := \{\widehat{\delta_i}\}_{i=1}^N \subset R^4$ inside the $S \times S$ local patches w.r.t. their center pixels, where $\widehat{\delta_i} = (\widehat{\delta x_i^A}, \widehat{\delta y_i^A}, \widehat{\delta x_i^B}, \widehat{\delta y_i^B})$. In the second head, *i.e.*, the classification head, we apply a sigmoid function to the outputs of the fc layer to obtain the confidence scores $\widehat{\mathcal{C}}_{pixel} = (\widehat{c_1}, \dots, \widehat{c_N}) \in R^N$, which express the validity of the detected matches. This allows us to detect and discard bad match proposals that cannot deliver a good pixel-wise match. We obtain the mid-level matches $\widehat{M}_{pixel} := \{\widehat{m_i}\}_{i=1}^N$ by adding the local matches to patch matches, *i.e.*, $\widehat{m_i} = m_i + \widehat{\delta_i}$. Features are collected again for the new set of local $S \times S$ patch pairs centered by the mid-level matches and fed into the fine-level regressor, which follows the same procedure as the mid-level regression to output the final pixel-level matches $\widetilde{M}_{pixel} := \{\widetilde{m_i}\}_{i=1}^N$ and the confidence scores $\widetilde{\mathcal{C}}_{pixel} = (\widetilde{c_1}, \dots, \widetilde{c_N}) \in R^N$.

3.2. Losses

Our pixel-level matching loss $\mathcal{L}_{pixel}$ involves two terms: (i) a classification loss $\mathcal{L}_{cls}$ for the confidence scores, trained to predict whether a match proposal contains a true match or not, and (ii) a geometric loss $\mathcal{L}_{geo}$ to judge the accuracy of the regressed matches. The final loss is defined as $\mathcal{L}_{pixel} = \alpha\mathcal{L}_{cls} + \mathcal{L}_{geo}$, where α is a weighting parameter to balance the two losses. We empirically set $\alpha = 10$ based on the magnitude of the two losses during training.

Sampson distance. To identify pixel-level matches, we supervise the network to find correspondences that agree with the epipolar geometry between an image pair. It defines that the two correctly matched points should lie on their corresponding epipolar lines when being projected to the other image using the relative camera pose transformation. How much a match prediction fulfills the epipolar geometry can be precisely measured by the Sampson distance. Given a match m_i and the fundamental matrix $F \in R^{3\times 3}$ computed by the relative camera pose of the image pair, its Sampson distance ϕ_i measures the geometric error of the match w.r.t. the fundamental matrix [11], which is defined as:

$$\phi_i = \frac{((P_i^B)^T F P_i^A)^2}{(FP_i^A)_1^2 + (FP_i^A)_2^2 + (F^T P_i^B)_1^2 + (F^T P_i^B)_2^2}, \quad (1)$$

where $P_i^A = (x_i^A, y_i^A, 1)^T, P_i^B = (x_i^B, y_i^B, 1)^T$ and $(FP_i^A)_k^2, (FP_i^B)_k^2$ represent the square of the k-th entry of the vector FP_i^A, FP_i^B.

Classification loss. Given a pair of patches obtained from a match proposal $m_i = (x_i^A, y_i^A, x_i^B, y_i^B)$, we label the pair as positive, hence define its classification label as $c_i^* = 1$, if $\phi_i < \theta_{cls}$. Here, θ_{cls} is our geometric distance threshold for classification. All the others pairs are labeled as negative. Given the set of predicted confidence scores $\mathcal{C}$ and the binary labels $\mathcal{C}^*$, we use the weighted binary cross entropy to measure the classification loss as

$$\mathcal{B}(\mathcal{C}, \mathcal{C}^*) = -\frac{1}{N}\sum_{i=1}^{N} w c_i^* \, log\, c_i + (1-c_i^*)\, log\,(1-c_i)\,, \quad (2)$$

where the weight $w = |\{c_i^* | c_i^* = 0\}| / |\{c_i^* | c_i^* = 1\}|$ is the factor to balance the amount of positive and negative patch pairs. We have separate thresholds $\widehat{\theta}_{cls}$ and $\widetilde{\theta}_{cls}$ used in the mid-level and the fine-level classification loss, which are summed to get the total classification loss $\mathcal{L}_{cls}$.

Geometric loss. To avoid training our regressors to refine matches within match proposals which are going to be classified as non-valid, for every refined match, we optimize its geometric loss only if the Sampson distance of its parent match proposal is within a certain threshold θ_{geo}. Our geometric loss is the average Sampson distance of the set of refined matches that we want to optimize. We use thresholds $\widehat{\theta}_{geo}$ and $\widetilde{\theta}_{geo}$ for the mid-level and the fine-level geometric loss accordingly and the sum of the two losses gives the total geometric loss $\mathcal{L}_{geo}$.

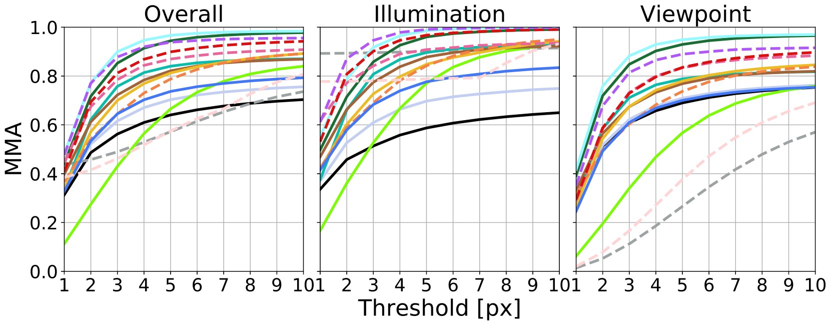
4. Implementation Details

We train *Patch2Pix* with match proposals detected by our adapted NCNet, *i.e.*, the pre-trained NC matching layer from [31], to match features extracted from our backbone. Our refinement network is trained on the large-scale outdoor dataset MegaDepth [14], where we construct 60661 matching pairs. We set the distance thresholds to compute the training losses (*c.f.* Sec. 3.2) as $\widehat{\theta}_{cls} = \widehat{\theta}_{geo} = 50$ for the mid-level regression and $\widetilde{\theta}_{cls} = \widetilde{\theta}_{geo} = 5$ for the fine-level regression. We constantly set the local patch size to $S = 16$ pixels at image resolution. The pixel-level matching is optimized using Adam [12] with an initial learning rate of $5e^{-4}$ for 5 epochs and then $1e^{-4}$ until it converges. A mini-batch input contains 4 pairs of images with resolution 480 × 320. We present architecture details about our regressor and our adapted NCNet [31], training data processing, hyper-parameter ablation, and qualitatively results of our matches in the supp. mat. (*c.f.* Sec. A & B).

5. Evaluation on Geometrical Tasks

5.1. Image Matching

As our first experiment, we evaluate *Patch2Pix* on the HPatches [1] sequences under the image matching task, where a method is supposed to detect correspondences between an input image pair. We follow the setup proposed in D2Net [6] and report the mean matching accuracy (MMA) [19] under thresholds varying from 1 to 10 pixels, together with the numbers of matches and features.



Methods	#Features / Matches
━ HesAff [18] + RootSIFT + NN	6.7K / 2.8K
━ HAN [21] + HN++ [20] + NN	3.9K / 2.0K
━ SuperPoint [5] + NN	2.0K / 1.1K
━ D2Net [6] + NN	6.0K / 2.5K
━ R2D2 [28] + NN	5.0K / 1.6K
━ ASLFeat [17] + NN	4.0K / 2.0K
━ SuperPoint + SuperGlue [33] (c=0.2)	0.5K
━ SuperPoint + SuperGlue [33] (c=0.9)	0.4K
━ SuperPoint + CAPS [40] + NN	2.0K / 1.1K
╍ SIFT [15] + CAPS + NN	4.4K / 1.5K
╍ DELF [25] + NN	4.6K / 1.9K
╍ SparseNCNet [30] (im3200, top2k)	2.0K
╍ NCNet([31] (Our Adapted)	1.5K
╍ Patch2Pix(c=0.5)	1.1K
╍ Patch2Pix(c=0.9)	0.7K

Figure 4. **Image Matching on HPatches [1].** We denote weakly-supervised methods with dashed lines and methods based on full supervision with solid lines.

Experimental setup. We use the confidence scores produced by the fine-level regressor to filter out outliers and study its performance under two settings, *i.e.*, $c = 0.5/0.9$, which present a trade-off between quantity and quality of the matches. To show the effectiveness of our refinement concept, we compare to our NCNet baseline, which provides our match proposals. For NCNet and *Patch2Pix*, we resize images to have a larger side of 1024 to reduce runtime. We also compare to SparseNCNet [30], which is the most similar one to ours among related works, since it also builds upon NCNet and aims to improve the accuracy of its matches through a re-localization mechanism. Besides comparing to several local feature methods that use NN Search for matching, we further consider SuperPoint [5] features matched with SuperGlue [33] and study its performance under their default threshold $c = 0.2$ and a higher threshold $c = 0.9$ for outlier rejection.

Results. As shown in Fig. 4, NCNet performs competitively for illumination sequences with constant viewpoints, which is a special case for NCNet since it uses fixed upsampling to bring patch matches to pixel correspondences. While its performance under illumination changes reveals its efficiency in patch-level matching, its accuracy under viewpoint changes reveals its insufficient pixel-level matching performance. Our refinement network brings patch-level matches predicted by NCNet to pixel-level correspondences, which drastically improves the matching accuracy under viewpoint changes and further improves under illumination changes. When comparing *Patch2Pix* to all weakly supervised methods, our model is the best at both thresholds under illumination changes. For viewpoint changes, our model with threshold $c = 0.9$ is the best and SparseNCNet performs similar to our model under threshold $c = 0.5$. Compared to the methods trained with full supervision, our model with threshold $c = 0.9$ outperforms all of them under illumination variations. For viewpoint changes, we are less accurate than SuperPoint + SuperGlue but still, we outperform all the other fully-supervised methods. Looking at the curves and the table in Fig. 4 together, both SuperPoint + SuperGlue and our method improve performance when using a higher threshold to remove less confident predictions.

5.2. Homography Estimation

Having accurate matches does not necessarily mean accurate geometry relations can be estimated from them since the distribution and number of matches are also important when estimating geometric relations. Therefore, we next evaluate *Patch2Pix* on the same HPatches [1] sequences for homography estimation.

Experimental setup. We follow the corner correctness metric used in [5, 33, 40] and report the percentage of correctly estimated homographies whose average corner error distance is below 1/3/5 pixels. In the following experiments, where geometrics relations are estimated using RANSAC-based solvers, we use $c = 0.25$ as our default confidence threshold, which overall gives us good performance across tasks. The intuition of setting a lower threshold is to filter out some very bad matches but leave as much information as possible for RANSAC to do its own outlier rejection. We compare to methods that are more competitive in the matching task which are categorized based on their supervision types: fully supervised (Full), weakly supervised (Weak), and mixed (Mix) if both types are used. We run all methods under our environment and measure the matching time from the input images to the output matches. We provide more experimental setup details in our supp. mat (*c.f*. Sec. C).

Results. From the results shown in Tab. 1, we observe again that NCNet performs extremely well under illumination changes due to their fixed upsampling (*c.f.* Sec. 5.2). Here, we verify that the improvement of matches by *Patch2Pix* under viewpoint changes is also reflected in the quality of the estimated homographies. Both SparseNCNet and our method are based on the concept of improving match accuracy by searching inside the matched local patches to progressively re-locate a more accurate match in higher resolution feature maps. While our method predicts matches at the original resolution and is fully learnable, their non-learning approach produces matches at a 4-times downscaled resolution. As we show in Tab. 1, our refinement network is more powerful than their re-localization

Method	Overall	Illumination	Viewpoint	Supervision	#Matches	Time (s)
	Accuracy (%, $\epsilon < 1/3/5$ px)					
SuperPoint [5] + NN	0.46 / 0.78 / 0.85	0.57 / 0.92 / 0.97	0.35 / 0.65 / 0.74	Full	1.1K	0.12
D2Net [6] + NN	0.38 / 0.72 / 0.81	0.65 / 0.95 / **0.98**	0.13 / 0.51 / 0.65	Full	2.5K	1.61
R2D2 [28] + NN	0.47 / 0.78 / 0.83	0.63 / 0.93 / **0.98**	0.33 / 0.64 / 0.70	Full	1.6K	2.34
ASLFeat [17] + NN	0.48 / 0.81 / 0.88	0.63 / 0.94 / **0.98**	0.34 / 0.69 / 0.78	Full	2.0K	0.66
SuperPoint + SuperGlue [33]	**0.51 / 0.83 / 0.89**	0.62 / 0.93 / **0.98**	**0.41/ 0.73/ 0.81**	Full	0.5K	0.14
SuperPoint + CAPS [40] + NN	0.49 / 0.79 / 0.86	0.62 / 0.93 / 0.98	0.36 / 0.65 / 0.75	Mix	1.1K	0.36
SIFT + CAPS [40] + NN	0.36 / 0.76 / 0.85	0.48 / 0.89 / 0.95	0.26 / 0.65 / 0.76	Weak	1.5K	0.73
SparseNCNet [30] (im3200, top2k)	0.36 / 0.66 / 0.76	0.62 / 0.92 / 0.97	0.13 / 0.41 / 0.57	Weak	2.0K	5.83
NCNet [31] (Our Adapted)	0.48 / 0.61 / 0.71	**0.98 / 0.98 / 0.98**	0.02 / 0.28 / 0.46	Weak	1.5K	0.83
Patch2Pix	**0.51** / 0.79 / 0.86	0.72 / 0.95 / **0.98**	0.32 / 0.64 / 0.75	Weak	1.3K	1.24
Oracle	0.00 / 0.15 / 0.54	0.00 / 0.23 / 0.7	0.00 / 0.07 / 0.39	-	2.5K	0.04
Patch2Pix (w.Oracle)	0.55 / 0.85 / 0.92	0.68 / 0.95 / 0.99	0.43 / 0.76 / 0.82	Weak	2.5K	0.76

Table 1. **Homography Estimation on Hpatches [1].** We report the percentage of correctly estimated homographies whose average corner error distance is below 1/3/5 pixels. We denote the supervision type with 'Full' for fully-supervised methods, 'Weak' for weakly-supervised ones, and 'Mix' for those used both types. We mark the best accuracy in **bold**.

mechanism, improving the overall accuracy within 1 pixel by 15 percent. For illumination changes, we are the second-best after NCNet, but we are better than all fully supervised methods. Under viewpoint variations, we are the best at 1-pixel error among weakly-supervised methods and we achieve very close overall accuracy to the best fully supervised method SuperPoint + SuperGlue.

Oracle Investigation. Since our method can filter out bad proposals but not generate new ones, our performance will suffer if NCNet fails to produce enough valid proposals, which might be the reason for our relatively lower performance on viewpoint changes. In order to test our hypothesis, we replace NCNet with an Oracle matcher to predict match proposals. Given a pair of images, our Oracle first random selects 2.5K matches from the GT correspondences computed using the GT homography and then randomly moves each point involved in a match within the 12 × 12 local patch centered at the GT location. In this way, we obtain our synthetic match proposals where we know there exists at least one GT correspondence inside the 16 × 16 local patches centered by those match proposals, which allows us to measure the performance of our true contribution, the refinement network. As shown in Tab. 1, the low accuracy of matches produced by our Oracle evidently verifies that the matching task left for our refinement network is still challenging. Our results are largely improved by using the Oracle proposals, which means our current refinement network is heavily limited by the performance of NCNet. Therefore, in the following localization experiments, to see the potential of our refinement network, we will also investigate the performance when using SuperPoint + SuperGlue to generate match proposals.

5.3. Outdoor Localization on Aachen Day-Night

We further show the potential of our approach by evaluating *Patch2Pix* on the Aachen Day-Night benchmark (v1.0) [34, 35] for outdoor localization under day-night illumination changes.

Method	Supervision	Localized Queries (%, 0.25m,2°/0.5m,5°/1.0m, 10°)	
		Day	Night
Local Feature Evaluation on Night-time Queries			
SuperPoint [5] + NN	Full	-	73.5 / 79.6 / 88.8
D2Net [6] + NN	Full	-	74.5 / 86.7 / **100.0**
R2D2 [28] + NN	Full	-	76.5 / **90.8** / **100.0**
SuperPoint + S2DNet [10]	Full	-	74.5 / 84.7 / **100.0**
ASLFeat [17] + NN	Full	-	77.6 / 89.8 / **100.0**
SuperPoint + CAPS [40] + NN	Mix	-	**82.7** / 87.8 / **100.0**
DualRC-Net [13]	Full	-	79.6 / 88.8 / 100.0
SIFT + CAPS [40] + NN	Weak	-	77.6 / 86.7 / 99.0
SparseNCNet [30]	Weak	-	76.5 / 84.7 / 98.0
Patch2Pix	Weak	-	79.6 / 87.8 / **100.0**
Full Localization with HLOC [32]			
SuperPoint [5] + NN	Full	85.4 / 93.3 / 97.2	75.5 / 86.7 / 92.9
SuperPoint + CAPS [40] + NN	Mix	86.3 / 93.0 / 95.9	83.7 / 90.8 / 96.9
SuperPoint + SuperGlue [33]	Full	**89.6** / 95.4 / **98.8**	86.7 / 93.9 / **100.0**
Patch2Pix	Weak	84.6 / 92.1 / 96.5	82.7 / 92.9 / 99.0
Patch2Pix (w.CAPS)	Mix	86.7 / 93.7 / 96.7	85.7 / 92.9 / 99.0
Patch2Pix (w.SuperGlue)	Mix	89.2 / **95.5** / 98.5	**87.8 / 94.9 / 100.0**

Table 2. **Evaluation on Aachen Day-Night Benchmark (v1.0) [34, 35].** We report the percentage of correctly localized queries under specific error thresholds. We follow the supervision notations described in Tab. 1 and mark the best results in **bold**.

Experimental Setup. To localize Aachen night-time queries, we follow the evaluation setup from the website[1]. For evaluation on day-time and night-time images together, we adopt the hierarchical localization pipeline (HLOC[2]) proposed in [32]. Matching methods are then plugged into the pipeline to estimate 2D correspondences. We report the percentage of correctly localized queries under specific error thresholds. We test our *Patch2Pix* model with NCNet proposals and SuperPoint [5] + SuperGlue [33] proposals. Note, the model has been only trained on NCNet proposals. Due to the triangulation stage inside the localization pipeline, we quantize our matches by representing keypoints that are closer than 4 pixels to each other with their mean location. We provide a more detailed discussion of the quantization inside our supp. mat (*c.f.* Sec. C).

Results. As shown in Tab. 2, for local feature evalua-

[1] https://github.com/tsattler/visuallocalizationbenchmark

[2] https://github.com/cvg/Hierarchical-Localization

Method	Supervision	Localized Queries (%, 0.25m/0.5m/1.0m, 10°) DUC1	 DUC2
SuperPoint [5] + NN	Full	40.4 / 58.1 / 69.7	42.0 / 58.8 / 69.5
D2Net [6] + NN	Full	38.4 / 56.1 / 71.2	37.4 / 55.0 / 64.9
R2D2 [28] + NN	Full	36.4 / 57.6 / 74.2	45.0 / 60.3 / 67.9
SuperPoint + SuperGlue [33]	Full	49.0 / **68.7** / 80.8	53.4 / 77.1 / **82.4**
SuperPoint + CAPS [40] + NN	Mix	40.9 / 60.6 / 72.7	43.5 / 58.8 / 68.7
SIFT + CAPS [40] + NN	Weak	38.4 / 56.6 / 70.7	35.1 / 48.9 / 58.8
SparseNCNet [30]	Weak	41.9 / 62.1 / 72.7	35.1 / 48.1 / 55.0
Patch2Pix	Weak	44.4 / 66.7 / 78.3	49.6 / 64.9 / 72.5
Patch2Pix (w.SuperPoint+CAPS)	Mix	42.4 / 62.6 / 76.3	43.5 / 61.1 / 71.0
Patch2Pix (w.SuperGlue)	Mix	**50.0** / 68.2 / **81.8**	**57.3 / 77.9** / 80.2

Table 3. **InLoc [39] Benchmark Results.** We report the percentage of correctly localized queries under specific error thresholds. Methods are evaluated inside the HLOC [32] pipeline to share the same retrieval pairs, RANSAC threshold, *etc*. We use the supervision notation from Tab. 1 and mark the best results in **bold**.

tion on night-time queries, we outperform the other two weakly-supervised methods. While being worse than SuperPoint [5] + CAPS [40], which involves both full and weak supervision, we are on-par or better than all the other fully-supervised methods. For full localization on all queries using HLOC, we show we are better than SuperPoint + NN on night queries and competitively on day-time images. By further substituting NCNet match proposals with SuperGlue proposals, we are competitive to SuperGlue on day-time images and outperform them slightly on night queries. Our intuition is that we benefit from our epipolar geometry supervision which learns potentially more general features without having any bias from the training data, which is further supported by our next experiment.

5.4. Indoor Localization on InLoc

Finally, we evaluate *Patch2Pix* on the InLoc benchmark [39] for large-scale indoor localization. The large textureless areas and repetitive structures present in its scenes makes this dataset very challenging.

Experimental Setup. Following SuperGlue [33], we evaluate a matching method by using their predicted correspondences inside HLOC for localization. We report the percentage of correctly localized queries under specific error thresholds. It is worth noting that compared to the evaluation on Aachen Day-Night, where our method looses accuracy up to 4 pixels due to the quantization, we have a fairer comparison on InLoc (where no triangulation is needed) to other methods. The results directly reflect the effect of our refinement when combined with other methods. Except for SuperPoint+SuperGlue, we evaluate several configurations of the other methods and compare to their best results. Please see the supp. mat. for more details (*c.f*. Sec. C).

Results. As shown in Tab. 3, *Patch2Pix* is the best among weakly supervised methods and outperforms all other methods except for SuperPoint + SuperGlue. Notice, we are 14.5 % better than SparseNCNet on DUC2 at the finest error, which further highlights that our learned refinement network is more effective than their hand-crafted relocalization mechanism. Further looking at the last rows of Tab. 3, our refinement network achieves the overall best performance among all methods when we replace NCNet proposals with more accurate proposals predicted by SuperPoint + SuperGlue. By searching inside the local regions of SuperPoint keypoints that are matched by SuperGlue, our network is able to detect more accurate and robust matches to outperform SuperPoint + SuperGlue. This implies that epipolar geometry is a promising type of supervision for the matching task. While CAPS is also trained with epipolar loss, its performance still largely relies on the keypoint detection stage. In contrast, we bypass the keypoint detection errors by working directly on the potential matches.

Generalization By evaluating *Patch2Pix* on image matching (*c.f*. Sec. 5.1) and homography estimation (*c.f*. Sec. 5.2), we validate our refinement concept by showing dramatic improvements over NCNet matches. While our network has been trained only on NCNet-type of proposals, we show that our refinement network provides distinct improvements, on both indoor and outdoor localization, by switching from the match proposals produced by NCNet to SuperPoint + SuperGlue proposals without the need for retraining. This highlights that our refinement network learns the general task of predicting matches from a pair of local patches, which works across different scene types and is independent of how the local patch pair has been obtained. Such general matching capability can be used to further improve the existing methods. As shown in Tab. 2 and Tab. 3, both SuperPoint + SuperGlue and SuperPoint + CAPS get improved by using our refinement network.

6. Conclusion

In this paper, we proposed a new paradigm to predict correspondences in a two-stage *detect-to-refine* manner, where the first stage focuses on capturing the semantic high-level information and the second stage focuses on the detailed structures inside local patches. To investigate the potential of this concept, we developed a novel refinement network, which leverages regression to directly output the locations of matches from CNN features and jointly predict confidence scores for outlier rejection. Our network was weakly supervised by epipolar geometry to detect geometrically consistent correspondences.We showed that our refinement network consistently improved our correspondence network baseline on a variety of geometry tasks. We further showed that our model trained with proposals predicted by a correspondence network generalizes well to other types of proposals during testing. By applying our refinement to the best fully-supervised method without retraining, we achieved state-of-the-art results on challenging long-term localization tasks.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, pages 5173–5182, 2017. 5, 6, 7

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3):346–359, 2008. 1

[3] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *ICCV*, pages 4322–4331, 2019. 1, 2, 3

[4] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, pages 2292–2300, 2013. 1, 2

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR Workshops*, pages 224–236, 2018. 1, 2, 3, 6, 7, 8

[6] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR*, 2019. 1, 2, 3, 5, 6, 7, 8

[7] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In *CVPR*, pages 253–262, 2019. 1, 2

[8] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *PAMI*, 2018. 1

[9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981. 2

[10] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2dnet: Learning accurate correspondences for sparse-to-dense feature matching. *ECCV*, 2020. 2, 3, 7

[11] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 5

[12] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 5

[13] Xinghui Li, Kai Han, Shuda Li, and Victor Adrian Prisacariu. Dual-resolution correspondence networks. In *NeurIPS*, 2020. 1, 2, 3, 7

[14] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 5

[15] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 6

[16] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *CVPR*, pages 2527–2536, 2019. 1

[17] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. In *CVPR*, pages 6589–6598, 2020. 1, 2, 3, 6, 7

[18] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004. 6

[19] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10):1615–1630, 2005. 5

[20] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NeurIPS*, pages 4826–4837, 2017. 6

[21] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *ECCV*, pages 284–300, 2018. 6

[22] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, pages 2666–2674, 2018. 1, 2, 3

[23] Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *PAMI*, 36(11):2227–2240, 2014. 2

[24] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *TRO*, 31(5):1147–1163, 2015. 1

[25] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, pages 3456–3465, 2017. 6

[26] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *NeurIPS*, pages 6234–6244, 2018. 2

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 2, 3

[28] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*, pages 12405–12415, 2019. 1, 2, 3, 6, 7, 8

[29] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, pages 6148–6157, 2017. 1, 2

[30] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. *ECCV*, 2020. 1, 2, 3, 6, 7, 8

[31] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2018. 1, 2, 3, 4, 5, 6, 7

[32] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 7, 8

[33] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 1, 2, 3, 6, 7, 8

[34] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 7

[35] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *British Machine Vision Conference (BMCV)*, 2012. 7

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[37] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *PJM*, 21(2):343–348, 1967. 1, 2

[38] Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. Acne: Attentive context normalization for robust permutation-equivariant learning. In *CVPR*, pages 11286–11295, 2020. 1, 2, 3

[39] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *CVPR*, 2018. 8

[40] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, 2020. 1, 2, 3, 6, 7, 8

[41] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, pages 127–134. IEEE, 2013. 1

[42] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. *ICCV*, 2019. 1, 2, 3

[43] Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. In *ICRA*, pages 3319–3326. IEEE, 2020. 2

7 Is Geometry Enough for Matching in Visual Localization?

7.1 Summary

In this paper, we propose a departure from the established paradigm of vision-based localization, which relies on the matching of visual descriptors between a query image and a 3D point cloud. While this approach yields high localization accuracy, it poses challenges in terms of storage requirements, privacy considerations, and long-term descriptor updates.

To elegantly address these practical concerns in large-scale localization, we introduce GoMatch, a novel approach that relies exclusively on geometric information for matching image keypoints to maps, represented as sets of bearing vectors. Our innovative representation of 3D points as bearing vectors significantly mitigates the cross-modal challenges in geometric-based matching that have hindered previous efforts in realistic environments.

Through careful architectural design, GoMatch surpasses prior work in geometric-based matching, achieving reductions of $(10.67m, 95.7°)$ and $(1.43m$, $34.7°)$ in average median pose errors on Cambridge Landmarks and 7-Scenes datasets, all while demanding only 1.5/1.7% of the storage capacity required by the leading visual-based matching methods.

Our findings affirm the potential and feasibility of real-world localization through geometric-based matching. We view our work as a foundational step in this emerging direction and anticipate that it will inspire further research in pursuit of even more accurate and dependable geometric-based visual localization systems. This marks a significant stride toward scalable, real-world visual localization solutions.

7.2 Author Contributions

The author of this dissertation significantly contributed to

- developing the main concepts
- implementing the algorithm
- evaluating the numerical experiments
- writing the paper

7.3 Preprint

Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is Geometry Enough for Matching in Visual Localization? In: *European Conference on Computer Vision*. Springer. 2022, pp. 407–425

Is Geometry Enough for Matching in Visual Localization?

Qunjie Zhou[1](✉), Sérgio Agostinho[2], Aljoša Ošep[1], and Laura Leal-Taixé[1]

[1] Technical University of Munich, Munich, Germany
{qunjie.zhou,aljosa.osep,leal.taixe}@tum.de
[2] Universidade de Lisboa, Lisbon, Portugal
sergio.agostinho@tecnico.ulisboa.pt
https://github.com/dvl-tum/gomatch

Abstract. In this paper, we propose to go beyond the well-established approach to vision-based localization that relies on visual descriptor matching between a query image and a 3D point cloud. While matching keypoints via visual descriptors makes localization highly accurate, it has significant storage demands, raises privacy concerns and requires update to the descriptors in the long-term. To elegantly address those practical challenges for large-scale localization, we present GoMatch, an alternative to *visual-based matching* that solely relies on geometric information for matching image keypoints to maps, represented as sets of bearing vectors. Our novel bearing vectors representation of 3D points, significantly relieves the cross-modal challenge in *geometric-based matching* that prevented prior work to tackle localization in a realistic environment. With additional careful architecture design, GoMatch improves over prior geometric-based matching work with a reduction of $(10.67\,\mathrm{m}, 95.7^\circ)$ and $(1.43\,\mathrm{m}, 34.7^\circ)$ in average median pose errors on Cambridge Landmarks and 7-Scenes, while requiring as little as 1.5/1.7% of storage capacity in comparison to the best visual-based matching methods. This confirms its potential and feasibility for real-world localization and opens the door to future efforts in advancing city-scale visual localization methods that do not require storing visual descriptors.

1 Introduction

In this paper we tackle scalable, data-driven visual localization. The ability to localize a query image within a 3D map based representation of the environment is vital in many applications, ranging from robotics to virtual and augmented reality. In past years, researchers have made a significant progress in

Q. Zhou and S. Agostinho—Equal contribution.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-20080-9_24.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13670, pp. 407–425, 2022.
https://doi.org/10.1007/978-3-031-20080-9_24

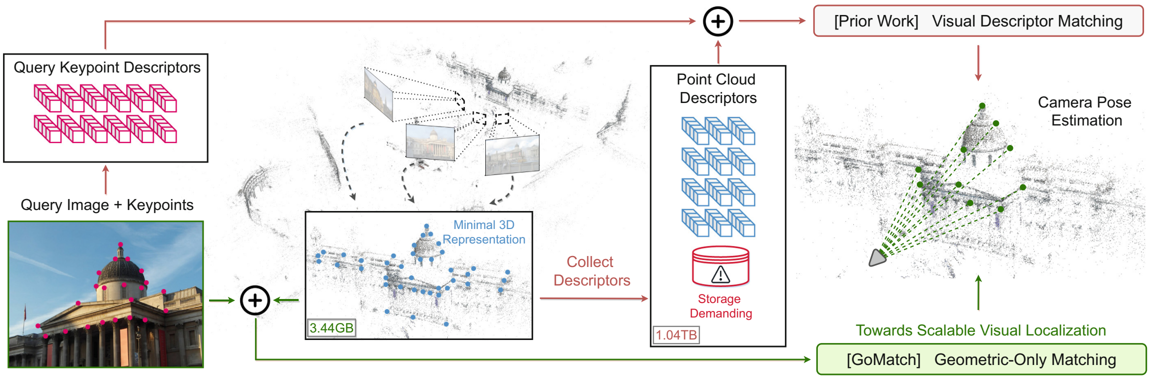

[Prior Work] Visual Descriptor Matching
Query Keypoint Descriptors
Query Image + Keypoints
Minimal 3D Representation
3.44GB
Collect Descriptors
Point Cloud Descriptors
Storage Demanding
1.04TB
Camera Pose Estimation
Towards Scalable Visual Localization
[GoMatch] Geometric-Only Matching

Fig. 1. In this work, we propose GoMatch to tackle visual localization w.r.t. a scene represented as a 3D point cloud. By relying only on geometric information for matching, GoMatch allows structure-based methods to achieve localization solely through the use of keypoints, sidestepping the need to store visual descriptors for matching. Keeping only the minimal representation of a 3D model, *i.e.*, its coordinates, leads to a more scalable pipeline towards large-scale localization that bypasses privacy concerns and is easy to maintain.

vision-based localisation [20,25,30,42,46,51,54,65,72,74]. The majority of methods [25,51,65,67,72] rely on a pre-built 3D representation of the environment, typically obtained using structure-from-motion (SfM) techniques [57,59]. Such 3D maps store 3D points and D-dimensional visual feature descriptors [55]. To determine the pose of a query image, *i.e.*, its 3D position and orientation, these methods match visual descriptors, obtained from the query image, with the ones stored in the point cloud. Once image-to-point-cloud matches are established, a Perspective-n-Point (PnP) solver [27,36] is used to estimate the camera pose. While working well in practice, this approach suffers from several drawbacks. First, we need to explicitly store per-point visual descriptors for point clouds, which hinders its applicability to large-scale environments due to the expensive storage requirement. Second, this limits the applicability to point clouds with specific descriptors, which increases the 3D map descriptor maintenance effort – maps need to be re-built or updated to be used in conjunction with newly developed descriptors [24]. Third, this approach in practice necessitates a visual descriptor exchange between the server (storing the 3D model and descriptors) and an online feature extractor. This is a point of privacy vulnerability, as human identities and personal information can be recovered from visual descriptors intercepted during the transmission [16,22,23,26,28,29,48,63]. The aforementioned issues lead to the main question we pose in this paper: *can we localize an image without relying on visual descriptors?* This would significantly reduce the map storage demands and get rid of descriptor maintenance. Recently, Campbell *et al.* [10,40] showed that it is feasible to directly match 2D image keypoints with a 3D point cloud using only geometrical cues. However, this is limited to ideal scenarios where outliers are not present. This assumption does not hold in real-world scenes and is not directly applicable to challenging visual localization. This is not surprising, as relying only on geometrical cues is a significantly more challenging compared to matching visual descriptors. In contrast to a single 2D/3D point coordinate, a visual descriptor provides a rich visual context, since it is commonly extracted from the local image patch centered around a keypoint [20,25,42,72] (Fig. 1).

In this paper, we achieve significant progress in making keypoints-to-point cloud direct matching ready for real-world visual localization. To cope with noisy images, point clouds, and inevitably keypoint outliers, we present **GoMatch**, a novel neural network architecture that relies on **G**eometrical information **o**nly. GoMatch leverages self- and cross- attention mechanisms to establish initial correspondences between image keypoints and point clouds, and further improves the matching robustness by filtering match outliers using a classifier. To the best of our knowledge, GoMatch is the first approach that is applicable to visual localization *in the wild* and does not rely on storage-demanding visual descriptors. In particular, compared to its prior work on geometric matching-based localization, GoMatch leads to a reduction of $(10.67\,\mathrm{m}, 95.7^\circ)$ and $(1.43\,\mathrm{m}, 34.7^\circ)$ in average median pose errors on Cambridge Landmarks dataset [35] and 7-Scenes dataset [61], confirming its potential in real-world visual localization.

We summarize our contributions as the following: (i) we develop a novel method to match query keypoints to a point cloud relying only on geometrical information; (ii) We bridge the difference in data modalities between a 2D image keypoint to a 3D point by representing it with its bearing vectors projected into co-visible reference views and show this is remarkably more robust compared to direct cross-modal matching; (iii) Our extensive evaluation shows that our method significantly outperforms prior work, effectively enabling real-world visual localization based on geometric-only matching; (iv) Finally, we thoroughly compare our method to the well-established visual localization baselines and discuss advantages and disadvantages of each approach. With this analysis, we hope to open the door for future progress towards more general and scalable structure-based methods for visual localization, which do not critically rely on storing visual descriptors, thereby reducing storage, relieving privacy concerns and eliminating the need for descriptor maintenance.

2 Related Work

Structure-Based Localization. Methods of this kind [5,50,53,58,66] commonly establish explicit correspondences between the query image pixels and the 3D points of the environment to compute the query image pose from the established matches using PnP solvers [27,36]. Keypoint correspondences are made by computing and matching visual descriptors for each keypoint from a query and database images [20,25,30,42,51,65,72]. Another recent work [52] iteratively optimizes a camera pose by minimizing visual descriptor distances between the 3D points observed in the query and the reference images. While it does not establish matches, it relies on visual descriptors extracted from a neural network and requires 3D points. Structure-based localization methods achieve impressive localization accuracy and state-of-the-art performance [20,50,51] in the long-term localization benchmark [54,67].

Table 1. On the challenges of large-scale structure-based localization. Analysis is performed on the MegaDepth [39] composed of many landmarks (similar to city districts), acting as an example of a city-scale dataset. We compare visual-based matching (VM) and geometric-based matching (GM) methods by analysing their storage requirement and considering whether a method requires to maintain map descriptors as well as provides privacy protection (*c.f.* the supplementary for more details.) For structured-based localization, scene coordinates (3D) and camera metadata (Cameras) are stored to obtain 2D-3D correspondences. In contrast to VM methods that need to additionally store visual descriptors or extract descriptors on-the-fly from the raw images, we show that using GM instead of VM, significantly reduces storage requirements, safeguards user privacy and bypasses the need for descriptor maintenance [24].

	Method	Desc. Maintenance	Privacy	Database Storage (GB, ↓)				Total
				Cameras (MB)	3D	Raw Ims	Descs	
VM	SIFT [41]	✗	✗	15.73	3.44	✗	130.10 (uint8)	133.33
	SuperPoint [20]	✗	✗	15.73	3.44	✗	1040.76 (fp32)	1044.21
	Extract on-the-fly	✗	✗	15.73	3.44	157.84	✗	161.29
Geometric-based Matching		✓	✓	15.73	3.44	✗	✗	**3.45**

Practical Challenges in Structure-Based Localization. Despite being highly accurate, modern localization solutions encounter practical challenges when deployed onto real-life applications, spanning city-level scale. The challenges are threefold: i) Relying on visual descriptors [20,25,42,72] makes the system demanding in storage[1] as shown in Table 1. To reduce storage requirement of the 3D scene representation, compression can be done by keeping a subset of the 3D points [13,14,43] and quantising [13,17,69] the descriptors associated with the 3D points. HybridSC [13] stands out among the existing work, with its extreme compression rate and minimal accuracy loss. ii) Localization methods following a server-client model need to transmit visual descriptors between the server and client, which exposes the model to a risk of a privacy breach [16,22,23,48]. To mitigate this issue, recent work [26,47] developed descriptors that are more robust against privacy attacks with slightly lower accuracy. iii) With the ongoing advancements in local features methods [20,25,26,42,47,72], continuously updating scene descriptors is a foreseeable demand [24] for visual-based matching methods. However, such an update requires either re-building the map with new descriptors or transforming the existing descriptors [24] to new ones. In this paper, we propose an *orthogonal* direction to address the storage, privacy and descriptor maintenance challenges in structure-based localization by relying solely on more lightweight geometric information for matching.

End-to-End Learned Localization. A recent trend of methods leverage data-driven techniques to learn to localize in an end-to-end manner, without relying on point clouds. This is achieved by either regressing scene coordinates, regressing the camera's absolute pose or regressing its relative pose w.r.t. to a database image. Scene coordinate regression methods [3,5,6,8,15,38,73] directly regress

[1] *Storage* as in non-volatile preservation of data, in contrast to volatile *memory*.

dense 3D scene coordinates from 2D images. However, they need to be re-trained for every new scene due to their lack of generalization [5–7,15]. In certain cases, multiple instances of the same network are trained on sub-regions of the scene, due to the limited capacity of a single network [7]. Therefore, it is unclear how to scale these methods [3,5,6,8,15,38,73], that are traditionally evaluated only on small indoor rooms, to large-scale scenes. Absolute pose regression (APR) methods implicitly encode the scene representation inside the network and directly regress the pose from the query image [33–35,49,71]. While earlier methods required training a model per scene and have been shown to overfit to the viewpoints and appearance of the training images [56], recent work in multi-scene APR [4,60] loosened the per-scene training requirements. Compared to multi-scene APR, our method generalizes across scenes as other structure-based localization methods (*c.f.* Sect. 5.5) while addressing its aforementioned practical challenges. Another related approach that sidesteps maintaining a 3D model with visual descriptors, is to regress relative camera poses [2,21,37,75] from a query image to its relevant database images. However, directly regressing the geometric transformations in general leads to limited generalization [56,75].

Direct Geometric Keypoint Matching. Matching image keypoints directly to 3D point clouds while jointly estimating pose has been widely investigated under relatively constrained environments [9–12,19,40,45]. Some require pose initialization [19] or pose distribution priors [45], while others, based on globally optimal estimators, have prohibitive runtime requirements in order to produce accurate estimates [9,11,12]. In contrast, the recent state-of-the-art, data-driven, geometric matching approaches [10,40] strike a good compromise between pose accuracy and time required to produce an accurate estimate. Despite not producing globally optimal solutions, BPnPNet [10] is able to estimate a reliable pose in a fraction of a second. Given a set of 2D keypoints in the query image and a set of 3D points in the scene point cloud, BPnPNet jointly estimates matches between these two sets *purely* based on geometric information. However, this approach was shown to work in idealistic scenarios assuming no outlier keypoints and, as we experimentally demonstrate, the matching performance degrades significantly once outliers are introduced. The outlier-free assumption clearly does not hold for challenging real-world localization scenarios as map building and keypoint detection are all challenging tasks, prone to errors and noise. In our work, we build upon BPnPNet and design a geometric matching module that is robust against keypoint outliers. We show in Sect. 5.3 that our approach significantly outperforms BPnPNet in matching keypoints with noisy outliers, effectively enabling the applicability of geometric-based matching to real-world visual localization.

3 Task Definition

Structure-Based Localization Pipeline. Structure-based methods assume as input a query image, a 3D point cloud of the scene, and database images with known poses. These methods first retrieve a set of database images that

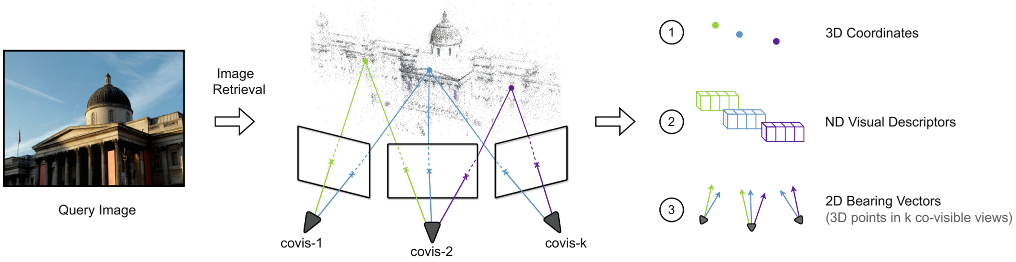


Fig. 2. Co-visible views & keypoint representations. Retrieving co-visible reference images (views) of a query image, narrows the matching against a full 3D point cloud to a subset of points that are more likely to be visible to the query image. Each 3D point can be represented differently by: 1) its 3D coordinate; 2) a visual descriptor that incorporates local appearance; or 3) a bearing vector that represents the direction from the reference camera origin to a 3D point in normalized coordinates. In this paper, we explore keypoint matching using representations 1) and 3).

are co-visible with the query image, *i.e.*, have a visual overlap, as illustrated in Fig. 2. Next, after narrowing down the search space, they establish 2D-3D correspondences between the query image keypoints and a (retrieved) subset of the 3D point cloud. This set of correspondences can be used to estimate the query image pose using a PnP solver [27,32]. The majority of prior work [20, 25,30,42,50,51,72] rely on storage-consuming visual descriptors, stored together with the point cloud, to establish 2D-3D matches. The key challenge we address is how to establish those correspondences *without* visual descriptors.

Problem Formulation. We assume two point sets, one with 2D keypoint coordinates in the image plane $\boldsymbol{p}_i \in \mathbb{R}^2$, and the second containing 3D point coordinates $\boldsymbol{q}_j \in \mathbb{R}^3$. We seek the matching set $\mathcal{M} := \{(i,j) | \boldsymbol{p}_i = \pi(\boldsymbol{q}_j; \mathtt{K}, \mathtt{R}, \boldsymbol{t})\}$, *i.e.*, the set of index pairs i and j, for which if the j-th 3D keypoint is projected to the image plane, it matches the coordinates specified by the corresponding i-th 2D point. The camera intrinsic matrix $\mathtt{K} \in \mathbb{R}^{3\times3}$ is assumed to be known, and the operator $\pi(\cdot)$ represents the camera projection function, which transforms 3D points onto the camera's frame of reference and projects them to the image plane according to the camera's intrinsics. Our goal is to find the correct 2D-3D keypoint matches for accurate pose estimation.

Keypoint Representation. We represent 2D pixels using 2D coordinates $(u, v) \in \mathbb{R}^2$ in the image plane. To learn a matching function that is agnostic to different camera models, we uplift those 2D points into a bearing vector representation $\boldsymbol{b} \in \mathbb{R}^2$, effectively removing the effect of the camera intrinsics. Bearing vectors encode the direction (or bearing) of points in a camera's frame of reference. We compute bearing vectors from image pixels as: $[\boldsymbol{b}^\top \; 1]^\top \propto \mathtt{K}^{-1}[u \, v \, 1]^\top$. For a 3D point, we consider two different representations (see Fig. 2): (i) as 3D coordinates $(x, y, z) \in \mathbb{R}^3$ w.r.t. a 3D world reference/origin; and (ii) as a bearing vector w.r.t. a reference database image. The bearing vector representation allows bringing both 2D pixels and 3D points to the same data modality. Given a 3D point $\boldsymbol{p} \in \mathbb{R}^3$ and transformation $(\mathtt{R}, \boldsymbol{t})$ from the world to the database image's frame of reference, we compute the corresponding bearing vector as:

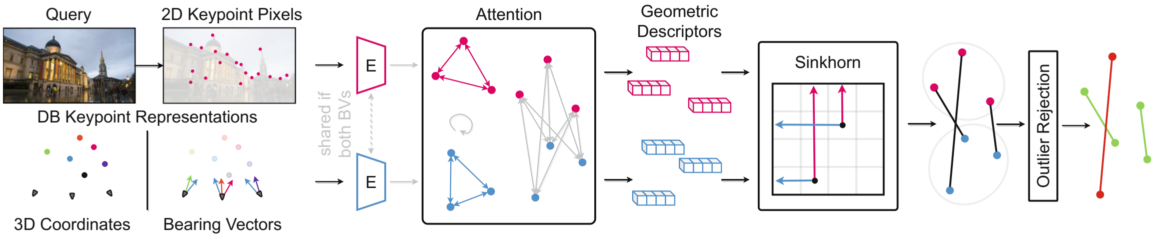


Fig. 3. GoMatch components overview. The query image and database keypoints first undergo a feature encoder **E** to generate per-point features. We share encoders in the query and database branch when database points are represented as bearing vectors otherwise not. These features are refined in the attention layer and then used in the Sinkhorn matching stage to establish an initial set of candidate matches, from which erroneous matches are filtered with an outlier rejection layer.

$$\boldsymbol{p}' = \mathtt{R}\boldsymbol{p} + \boldsymbol{t}, \;\; [\boldsymbol{b}^\top \; 1]^\top = \boldsymbol{p}'/p'_z, \tag{1}$$

where $\boldsymbol{p}'$ represents $\boldsymbol{p}$ in the camera's frame of reference, and $\boldsymbol{p}'_z$ represents its z coordinate. As shown in Table 1, these geometric-based point representations require significantly lower storage compared to visual descriptor based ones, *e.g.*, as low as 3% compared to the storage of modern descriptors.

4 Geometric-Only Matching

BPnPNet in a Nutshell. BPnPNet [10] made great progress towards establishing correspondence between the query keypoints and 3D point cloud in the absence of visual descriptors. It proposes an end-to-end trainable, differentiable matcher that performs 2D to 3D cross modal matching without relying on appearance information. While this is a step in the right direction, we show in Sect. 5.3 that it does not scale to the real-world visual localization scenarios where outliers, *i.e.* points without a match, are pervasive. Direct 2D-3D matching of sparse keypoints is a challenging problem due to low amount of discriminative data, *i.e.* points no longer have a local visual appearance, and its cross-modal nature. In a nutshell, BPnPNet (i) encodes points to obtain per point features, (ii) establishes matches using the Sinkhorn algorithm [18,62], which finds the optimal assignment between geometrical features, and finally, (iii) leverages a differentiable PnP solver that imposes an additional pose supervision on the network. In the following, we build on the observation that the lightweight geometric feature encoder does not possess the necessary representational power to produce features that generalize simultaneously to situations with and without outliers.

4.1 GoMatch: Embracing Outliers

In GoMatch we (i) propose architectural changes that enable resilience to outliers and (ii) cast the *cross-modal* nature of 2D-3D matching to an *intra-modal* setting through the use of bearing vectors. Below, we explain the details of these contributions, which are experimentally validated to be all necessary and critical to outlier-robust geometric matching in Sect. 5.3. We refer to Fig. 3 for a visual overview of the entire network. Furthermore, we add an outlier rejection layer to retain only quality matches from the Sinkhorn outputs. While we introduce the novel network components in the following paragraphs, we refer the reader to the supplementary material for an in-depth description of all network components.

Feature Refinement Through Attention. In BPnPNet, each keypoint node is processed in parallel with an MLP-style encoder to extract features directly for matching, and information exchange happens only in the Sinkhorn matching stage. This might lead to a learned feature representation which lacks context information within each 2D/3D modality and cross modality. Based on this assumption, we explore adding information exchange prior to matching. To enhance the context information within each modality, we apply *self-attention* to the raw encoded features where a graph neural network [31] refines features of every keypoint by exchanging the information with a fixed number of closest neighbors in coordinate space. This is followed by *cross-attention* [70], where every keypoint from one modality will interact with all keypoints from the other modality through a sequence of multi-head attention layers. By stacking several blocks of such self-/cross-attention layers, we are able to learn more representative features, which allows Sinkhorn to identify significantly better outlier matches.

Outlier Rejection. After Sinkhorn matching, the estimated corresponding pairs may still contain outlier matches. To filter those, we follow [44] and add a classifier that takes in the concatenated geometric features from the query and database keypoints, and predicts confidence scores for all matches. Estimated correspondences with confidence below a threshold (0.5 in practice) are rejected.

Matching with Bearing Vectors. Directly matching 2D keypoints to cross-modal 3D coordinates is challenging because it requires the network to learn features that have to consider not only the relationship between keypoints, but also the influence of different camera poses. Furthermore, the different distributions of 3D point clouds between datasets, *e.g.*, different scene sizes or different gravity directions, are particularly challenging for a single encoder to learn. Based on this observation, we propose to leverage the *bearing vector* representation of the database points to sidestep the difference in data modalities. In addition to nullifying the effects of the camera intrinsics, projecting 3D points as bearing vectors onto a "covisible" frame that is closer to the query frame (compared to the world reference frame), effectively mitigates the influence of the camera pose (viewpoint changes) during matching, albeit dependent on the quality of retrieval. Finally, bearing vectors provide a common modality between query and database keypoints, eliminating the need for a separate encoder. As we demonstrate in our experimental section, the change in input type has a substantial positive effect.

4.2 Training GoMatch

All of our models are trained to learn feature matching and outlier filtering jointly, using a matching loss and an outlier rejection loss.

Matching Loss. The Sinkhorn matching layer is trained to output a discrete joint probability distribution of two sets of keypoints being matched. We denote this distribution as $\tilde{\mathtt{P}} \in \mathbb{R}_{+}^{M+1 \times N+1}$, such that $\sum_{i=1}^{M+1} \sum_{j=1}^{N+1} \tilde{\mathtt{P}}_{ij} = 1$, *i.e.*, is a valid probability distribution. Here, M and N denote the total number of query and database keypoints considered during the matching. We include an extra row and column to allow keypoints not to be matched. We employ a negative log loss to the joint discrete probability distribution. Consider the set of all ground truth matches $\mathcal{M}$, as well as the set of unmatched query keypoints $\mathcal{U}_q$ and database keypoints $\mathcal{U}_d$. The matching loss is of the form:

$$L_{\text{match}} = -\frac{1}{N_m} \left(\sum_{(i,j) \in \mathcal{M}} \log \tilde{\mathtt{P}}_{ij} + \sum_{i \in \mathcal{U}_q} \log \tilde{\mathtt{P}}_{i(N+1)} + \sum_{j \in \mathcal{U}_d} \log \tilde{\mathtt{P}}_{(M+1)j} \right), \tag{2}$$

where $N_m = |\mathcal{M}| + |\mathcal{U}_q| + |\mathcal{U}_d|$.

Outlier Rejection Loss. For the outlier rejection layer we employ a mean weighted binary cross-entropy loss:

$$L_{\text{or}} = -\frac{1}{N_c} \sum_{i=1}^{N_c} w_i \left(y_i \log p_i + (1 - y_i) \log(1 - p_i) \right), \tag{3}$$

where N_c denotes the total number of correspondences supplied to the outlier rejection layer. The term p_i denotes the classifier output probability for each correspondence, while y_i denotes the correspondence target label, and w_i is the weight balancing the negative and positive samples. Our final loss balances both terms equally, *i.e.*, $L_{\text{total}} = L_{\text{match}} + L_{\text{or}}$. We present implementation details about training and testing process in our supplementary material.

5 Experimental Evaluation

In this section, we thoroughly study the potential of using our proposed geometric-based matching for the task of real-world visual localization. We start our experiments by testing the robustness of BPnPNet [10] and GoMatch with keypoint outliers. Next, we verify our technical contribution of successfully diagnosing the missing components leading to robust geometric matching and enabling geometry-based visual localization. Furthermore, we position gemetric-based localization among other state-of-the-art visual localization approaches by comprehensively analysing each method in terms of localization accuracy, descriptor maintenance effort [24], privacy risk, and storage demands (Sect. 5.4). Finally, we present a generalization study (Sect. 5.5) to highlight that our proposed method generalizes across different types of datasets and keypoint detectors. We hope that our in-depth study serves as a starting point of this rarely explored new direction, and inspires new work to advance scalable visual localization through geometric-only matching in the future.

5.1 Datasets

We use MegaDepth [39] for training and ablations, given its large scale. It consists of images captured in-the-wild from 196 outdoor landmarks. We adopt the original test set proposed in [39], and split the remaining sequences into training and validation sets. After verifying our best models on Megadepth, we evaluate them on the popular Cambridge Landmarks [35] (Cambridge) dataset which consists of 4 outdoor scenes of different scales. It allows for convenient comparison to other localization approaches. We use the reconstructions released by [52]. In addition, we evaluate on the indoor 7-Scenes [61] dataset to further assess the generalization capability of our method. 7-Scenes is composed of dense point clouds captured by an RGB-D sensor, and thus provides an alternative environment with different keypoint distributions, in both 2D images and 3D point clouds. We perform evaluation on the official test splits released by the Cambridge and 7-Scenes datasets. We provide detailed information about training data generation using MegaDepth in the supplementary.

5.2 Experimental Setup

Keypoint Detection. For MegaDepth and Cambridge, we use respectively SIFT [41] and SuperPoint [20], preserving the same keypoint detector used to reconstruct their 3D models. For 7-Scenes, we use both SIFT and SuperPoint to extract keypoints for both 2D images and 3D point cloud given RGB-D images.

Retrieval Pairs. We use ground truth to sample retrieval pairs that have at least 35% visual overlap in MegaDepth to ensure enough matches are present during training, as well as to isolate the side-effect of retrieval performance during ablations. For evaluation and comparison to state-of-the-art localization methods, we follow [52] and use their *top-10* pairs retrieved using NetVLAD [1] on Cambridge and DenseVLAD [68] on 7-Scenes.

Matching Baselines. We consider BPnPNet [10] as our geometric-based matching baseline. For a fair comparison, we re-train BPnPNet using our training data. Our visual-based matching baselines use SIFT [41] and SuperPoint [20] (SP) as keypoint descriptors. To match visual descriptors, we use nearest neighbor search [46] with mutual consistency by default and SuperGlue [51] (SG).

Localization Pipeline. Following the state-of-the-art structure-based localization, *e.g.*, HLoc [50], we first obtain up to $k = 10$ retrieval pairs between a query and database images. Then we establish per-pair 2D to 3D matches using either a geometric-based or a visual-based matching model, and then merge results from k pairs based on their matching scores to estimate camera poses. For fairness, all matching baselines use identical retrieval pairs and identical settings for the PnP+RANSAC solver [32].

Evaluation Metrics. For MegaDepth, we follow BPnPNet [10] to report the pose error quantiles at 25/50/75% for the translation and rotation (°) errors as evaluation metrics. However, as the scale unit of MegaDepth is undetermined and

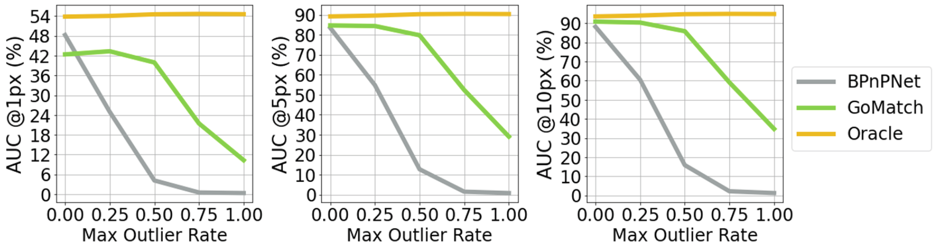


Fig. 4. Influence of keypoint outlier rate. In contrast to prior work BPnPnet [10], GoMatch is significantly more robust against keypoint outliers thanks to the more powerful attention-based architecture as well as our novel formulation of matching bearing vectors instead of cross-modal features.

varies between scenes, the translation errors are not consistent between scenes. Therefore, we propose a new metric based on pixel-level reprojection errors that preserves scene consistency. For each query, we project its inlier 3D keypoints using the predicted and the ground-truth poses. We then report the area under the cumulative curve (AUC) of the mean reprojection error up to 1/5/10px, inspired by the pose error based AUC metric used in [52,64]. We report the commonly used median translation (m) and rotation (°) errors [13,35,56] per-scene on Cambridge and 7-Scenes.

5.3 Ablations

We perform ablation studies with MegaDepth's [39] test split, where all retrieval pairs have guaranteed 35% co-visibility, to focus purely on matching performance. In addition, we study the effect of using a single co-visible reference view ($k = 1$) as a minimal setting, as well as multiple views, *e.g.*, $k = 10$, following the common practice in hierarchical structure-based localization [52,56]. To better understand the new AUC metric, we also present an **Oracle** that uses ground truth matches as its prediction. It is used to show the upper-bound performance that can be achieved using our metric and generated data.

Sensitivity to Keypoint Outliers. In a real-world localization setting, the detected query image keypoints will often be noisy and will not have a direct correspondence in the 3D point cloud. Keypoint matching methods thus need to be able to cope with outliers. We first study whether our baseline has this capability by manually increasing the maximum outlier rate, ranging from 0 to 1. The outlier rate is computed as the number of keypoints without a match divided by the total number of keypoints, taking the maximum between 2D and 3D. For all other experiments, we do not control keypoint the outlier rate to properly mimic realistic conditions. As shown in Fig. 4, the Oracle stays round 55/90/94% (AUC@1/5/10px). The large error at 1px is due to our match generation process (*c.f.* supplementary for a detailed discussion). BPnPNet [10] slightly outperforms GoMatch at 1px threshold, being similarly accurate to us at 5/10px thresholds in the absence of outliers. However, as the ratio of outliers increases, the performance of BPnPNet drastically drops, while GoMatch gracefully handles outliers,

Table 2. GoMatch ablation. *Top:* We present Oracle for reference and re-trained BPnPNet [10] as our baseline. *Middle:* We study how the 3D representation (Repr.) and architectural changes influences the performance. Using bearing vector (BVs) instead of 3D coordinates (Coords) as representation and introducing feature attention (Att) are the most crucial factors to the performance improvement. Together with further benefits from the outlier rejection (OR) component and sharing the query and database keypoint feature encoders leads us to the full GoMatch model (*Bottom*). All results rely on a singe retrieval image unless stated otherwise, *e.g.*, $k = 10$.

Model	3D	Share	Att	OR	Rotation (°)	Translation	Reproj. AUC (%)
	Repr.	Encoder			Quantile@25/50/75% (↓)		@1/5/10px (↑)
Oracle					0.03/0.06/0.10	0.00/0.00/0.01	54.58/90.37/94.87
BPnPNet	Coords	✗	✗	✗	15.17/31.05/59.78	1.67/3.14/5.31	0.34/0.83/1.21
BPnPNet ($k = 10$)	Coords	✗	✗	✗	16.03/33.27/63.90	1.59/3.24/5.80	0.56/1.08/1.50
Variants	BVs	✗	✗	✗	12.19/27.68/58.22	1.26/2.8/5.14	0.37/1.48/2.18
	BVs	✓	✗	✗	9.16/22.62/53.20	0.98/2.38/4.72	0.85/3.09/4.36
	BVs	✓	✓	✗	0.55/8.08/29.34	0.05/0.84/3.34	9.13/25.71/31.65
	BVs	✗	✓	✓	0.38/7.46/31.75	0.04/0.83/3.73	10.22/28.17/33.69
	Coords	✗	✓	✓	4.09/23.56/63.21	0.37/2.53/5.93	3.81/13.54/17.46
GoMatch	BVs	✓	✓	✓	0.36/6.97/29.85	0.03/0.69/3.38	10.30/29.08/34.79
GoMatch($k = 10$)	BVs	✓	✓	✓	**0.15/0.95/13.00**	**0.01/0.09/1.55**	**15.14/42.39/51.24**

i.e., GoMatch is always above 80% at 5/10px up to 50% of outliers. This experiment confirms that GoMatch is significantly more robust to outliers compared to BPnPNet. This outlier robustness is achieved through careful modifications to the network architecture and 3D point representation, both validated by a thorough performance analysis presented in the next sections.

Architecture-Level Analysis. In Table 2 (*Top*), we present the Oracle and BPnPNet [10] re-trained on our data for a direct comparison with GoMatch. This is paired with additional variants, progressively transitioning from BPnPNet to GoMatch. We found that shared encoding brings performance gains up to 0.48/1.61/2.18 AUC percentage points. Adding feature attention on top leads to a significant improvement of 8.28/22.62/27.29 AUC percentage points. By further adding the outlier rejection increases the AUC by 1.17/3.37/3.14% points. We conclude that these network components yield 9.93/27.6/32.61% points of improvements in terms of AUC scores when using bearing vectors the representation.

Representation-Level Analysis. Using 3D coordinates (Coords) instead of bearing vectors (BVs), even with attention and outlier rejection, hinders performance dramatically by 6.49/15.54/17.33% points. If we only change the representation from Coords to BVs, without attention nor outlier rejection, the improvement is merely 0.31/1.29/1.9% points. Therefore, we verify the bearing vector representation is as important as the architectural changes, and both contribute towards keypoint outlier resilience. By modifying both architecture and repre-

Table 3. Comparison to existing localization baselines. We consider end-to-end (E2E) methods and structure-based methods that either matches visual descriptors (VM) or geometries (GM). We report median translation and angular error for each landmark and combined storage requirements for operating on all landmarks. *No Desc. Maint.* is checked if a method does not require descriptor updates in the long run. *Privacy* is checked if a method is resilient to existing known privacy attacks.

	Method	Storage (MB)	No Desc. Maint.	Privacy	King's College	Old Hospital	Shop Facade	St. Mary's Church
					Median Pose Error (m, °) (↓)			
E2E	PoseNet [35]	200	✓	✓	1.92/5.40	2.31/5.38	1.46/8.08	2.65/8.48
	DSAC++ [6]	828	✓	✓	0.18/0.30	0.20/0.30	0.06/0.30	0.13/0.40
	MSPN [4]	-	✓	✓	1.73/3.65	2.55/4.05	2.92/7.49	2.67/6.18
	MS-Transformer [60]	71.1	✓	✓	0.83/1.47	1.81/2.39	0.86/3.07	1.62/3.99
VM	HybridSC [13]	3.13	✗	?	0.81/0.59	0.75/1.01	0.19/0.54	0.50/0.49
	Active Search [53]	812.7	✗	✗	0.42/0.55	0.44/1.01	0.12/0.40	0.19/0.54
	HLoc [50](w.SP [20])	3214.84	✗	✗	0.16/0.38	0.33/1.04	0.07/0.54	0.16/0.54
	HLoc(w.SP+SG [51])	3214.84	✗	✗	**0.12/0.20**	**0.15/0.30**	**0.04/0.20**	**0.07/0.21**
GM	BPnPNet [10]	48.15	✓	✓	26.73/106.99	24.8/162.99	7.53/107.17	11.11/49.74
	GoMatch	48.15	✓	✓	0.25/0.64	2.83/8.14	0.48/4.77	3.35/9.94

sentation, GoMatch outperforms the re-trained BPnPNet by 9.96/28.25/33.58 AUC percentage points.

Utilizing Multiple Co-visible Images. As shown in Table 2, when using $k = 10$ co-visible views, both methods improved their result: BPnPNet by a small margin and GoMatch by a large margin of 4.84/13.31/16.45 AUC percentage points. We thus use $k = 10$ for all of the following experiments.

5.4 Comparison to Localization Baselines

Following the discussion in Sect. 2, we comprehensively compare GoMatch with other established baselines by looking beyond localization performance, and considering as well the storage footprint, resiliency to privacy attacks, and descriptor maintenance. As shown in Table 3, HLoc with SuperPoint and SuperGlue is the most accurate method but also has the highest storage requirements while being vulnerable to privacy attacks. Using HLoc with a newly developed descriptor method will require the map to be updated. In end-to-end methods, DSAC++ is the most accurate method while being resilient to privacy attacks as it does not need to transmit visual descriptors. However, as it requires 4 model versions trained per-scene, it requires 828 MB storage to work under 4 scenes compared to our 48.12 MB. HybridSC as the most storage-efficient method keeps only 1.5% if its original points via compression. However, it is unclear whether the privacy issue still remains for this method since it still relies on full visual descriptors to perform matching. Notice, compressing scene structure can be theoretically combined with GoMatch to lower our storage requirements, which we leave as future work to design suitable scene compression techniques for geometric-base matching. On the whole, GoMatch and MS-Transformer both properly balance those three aspects showing benefits in storage, privacy and absence of descriptor maintenance, and are competitive in accuracy. Compared to its visual-descriptor

Table 4. Generalization study on 7-Scenes. GoMatch generalizes between different scene types and detector types and outperforming BPnPNet and PoseNet.

	Method	Storage (MB)	No Desc. Maint.	Privacy	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
					Median Pose Error (m, °) (↓)						
E2E	PoseNet [35]	350	✓	✓	0.32/8.12	0.47/14.4	0.29/12.0	0.48/7.68	0.47/8.42	0.59/8.64	0.47/13.8
	DSAC++ [6]	1449	✓	✓	**0.02/0.50**	**0.02/0.90**	**0.01**/0.80	**0.03/0.70**	**0.04/1.10**	**0.04/1.10**	0.09/2.60
	MSPN [4]	-	✓	✓	0.09/4.76	0.29/10.5	0.16/13.1	0.16/6.8	0.19/5.5	0.21/6.61	0.31/11.63
	MS-Transformer [60]	71.1	✓	✓	0.11/4.66	0.24/9.6	0.14/12.19	0.17/5.66	0.18/4.44	0.17/5.94	0.26/8.45
VM	Active Search [53]	-	✗	✗	0.04/1.96	0.03/1.53	0.02/1.45	0.09/3.61	0.08/3.10	0.07/3.37	**0.03**/2.22
	HLoc [50](w.SIFT [41])	2923	✗	✗	0.03/1.13	0.03/1.08	0.02/2.19	0.05/1.42	0.07/1.80	0.06/1.84	0.18/4.41
	HLoc(w.SP [20])	22977	✗	✗	0.03/1.28	0.03/1.3	0.02/1.99	0.04/1.31	0.06/1.63	0.06/1.73	0.07/1.91
	HLoc(w.SP+SG [51])	22977	✗	✗	**0.02**/0.85	**0.02**/0.94	**0.01/0.75**	**0.03**/0.92	0.05/1.30	**0.04**/1.40	0.05/**1.47**
GM	BPnPNet [10](SIFT [41])	302	✓	✓	1.29/43.82	1.48/51.82	0.93/55.13	2.61/59.06	2.15/39.85	2.15/43.00	2.98/60.27
	BPnPNet (SP [20])	397	✓	✓	1.25/43.9	1.42/45.09	0.8/50.05	2.33/14.54	1.71/31.81	1.68/33.91	2.1/55.78
	GoMatch (SIFT)	302	✓	✓	0.04/1.65	0.13/3.86	0.09/5.17	0.11/2.48	0.16/3.32	0.13/2.84	0.89/21.12
	GoMatch (SP)	397	✓	✓	0.04/1.56	0.12/3.71	0.05/3.43	0.07/1.76	0.28/5.65	0.14/3.03	0.58/13.12

SuperPoint counterpart, GoMatch requires only 1.5% of the capacity to store same scene. GoMatch reduces the average pose errors by $(10.67\,\mathrm{m}, 95.7^\circ)$ compared to our only prior geometric-based matching work, significantly reducing the accuracy gap to state-of-the-art methods. We hope this inspires researchers to pursue this line of work.

5.5 Generalization

As our final experiment, we study the generalization capability of our method in terms of localization in different types of scenes, *e.g.*, indoor and outdoor, and matching keypoints obtained using different detectors. According to our results in Table 4, similar to our previous experiments, we outperform BPnPNet by a large margin achieving $(1.43\,\mathrm{m}, 34.7^\circ)$ lower average median pose errors. Except for GoMatch with SIFT keypoints which produces a relatively large 21.12° median rotation error in Stairs, we are only slightly worse than our visual-based matching baselines with SIFT and SuperPoint. Yet, we require only 10/1.7% of the storage that is required by SIFT/SuperPoint to store maps. We also largely outperform PoseNet [35] in all metrics for all scenes except for the relatively lower translation error in Stairs scene, *i.e.*, $(0.47\,\mathrm{m}$ vs $0.58\,\mathrm{m})$. Furthermore, we achieve better pose than MS-Transformer in the majority of scenes, at the expense of a higher storage requirement. The results clearly verify that GoMatch trained on outdoor scenes (MegaDepth) generalizes smoothly to indoor scenes (7-Scenes), being agnostic to scene types. Similarly, we also confirm that GoMatch trained with SIFT keypoints generalizes well to SuperPoint keypoints, being agnostic to detector types.

6 Conclusion

We present GoMatch, a novel sparse keypoint matching method for visual localization that relies only on geometrical information and that carefully balances common practical challenges of large-scale localization, namely: localization performance, storage demands, privacy and descriptor maintenance (or lack thereof). From all these, the last three are often overlooked. Through a rigorous architecture design process, GoMatch dramatically surpasses its prior work in handling outliers, enabling it for real-world localization. Compared to localization pipelines using visual descriptor-based matching, GoMatch allows localization with a minimal 3D scene representation, requiring as little as 1.5/1.7% to store the same scene. Geometric-based matching brings localization pipelines to a new level of scalability that opens the door for localizing in much larger environments. We see our work as a starting point for this new direction and we look forward to inspire other researchers to pursue more accurate and reliable geometric-based visual localization in the future.

Acknowledgments. This research was partially funded by the Humboldt Foundation through the Sofja Kovalevskaya Award.

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
2. Balntas, V., Li, S., Prisacariu, V.: RelocNet: continuous metric learning relocalisation using neural nets. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 782–799. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_46
3. Bhowmik, A., Gumhold, S., Rother, C., Brachmann, E.: Reinforced feature points: optimizing feature detection and description for a high-level task. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4948–4957 (2020)
4. Blanton, H., Greenwell, C., Workman, S., Jacobs, N.: Extending absolute pose regression to multiple scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2020)
5. Brachmann, E., et al.: DSAC - differentiable RANSAC for camera localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
6. Brachmann, E., Rother, C.: Learning less is more - 6D camera localization via 3D surface regression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
7. Brachmann, E., Rother, C.: Expert sample consensus applied to camera relocalization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7525–7534 (2019)
8. Brachmann, E., Rother, C.: Neural-guided RANSAC: learning where to sample model hypotheses. In: IEEE International Conference on Computer Vision (ICCV), pp. 4322–4331 (2019)
9. Brown, M., Windridge, D., Guillemaut, J.-Y.: Globally optimal 2D-3D registration from points or lines without correspondences. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
10. Campbell, D., Liu, L., Gould, S.: Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 244–261. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_15

11. Campbell, D., Petersson, L., Kneip, L., Li, H.: Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
12. Campbell, D., Petersson, L., Kneip, L., Li, H., Gould, S.: The alignment of the spheres: globally-optimal spherical mixture alignment for camera pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
13. Camposeco, F., Cohen, A., Pollefeys, M., Sattler, T.: Hybrid scene compression for visual localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
14. Cao, S., Snavely, N.: Minimal scene descriptions from structure from motion models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
15. Cavallari, T., Bertinetto, L., Mukhoti, J., Torr, P., Golodetz, S.: Let's take this online: adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In: 2019 International Conference on 3D Vision (3DV), pp. 564–573 (2019)
16. Chelani, K., Kahl, F., Sattler, T.: How privacy-preserving are line clouds? Recovering scene details from 3D lines. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15668–15678 (2021)
17. Cheng, W., Lin, W., Chen, K., Zhang, X.: Cascaded parallel filtering for memory-efficient image-based localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
18. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. In: Burges, C.J., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 26. Curran Associates Inc. (2013)
19. David, P., Dementhon, D., Duraiswami, R., Samet, H.: SoftPOSIT: simultaneous pose and correspondence determination. Int. J. Comput. Vis. **59**(3), 259–284 (2004)
20. DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: self-supervised interest point detection and description. In: CVPR Workshops, pp. 224–236 (2018)
21. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: CamNet: coarse-to-fine retrieval for camera re-localization. In: IEEE International Conference on Computer Vision (ICCV), pp. 2871–2880 (2019)
22. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 29. Curran Associates Inc. (2016)
23. Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4829–4837 (2016)
24. Dusmanu, M., Miksik, O., Schonberger, J.L., Pollefeys, M.: Cross-descriptor visual localization and mapping. In: IEEE International Conference on Computer Vision (ICCV), pp. 6058–6067 (2021)
25. Dusmanu, M., et al.: D2-Net: a trainable CNN for joint detection and description of local features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
26. Dusmanu, M., Schönberger, J.L., Sinha, S.N., Pollefeys, M.: Privacy-preserving image features via adversarial affine subspace embeddings. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

27. Gao, X.-S., Hou, X.-R., Tang, J., Cheng, H.-F.: Complete solution classification for the perspective-three-point problem. IEEE Trans. Pattern Anal. Mach. Intell. **25**(8), 930–943 (2003)
28. Geppert, M., Larsson, V., Speciale, P., Schönberger, J.L., Pollefeys, M.: Privacy preserving structure-from-motion. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 333–350. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_20
29. Geppert, M., Larsson, V., Speciale, P., Schonberger, J.L., Pollefeys, M.: Privacy preserving localization and mapping from uncalibrated cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1809–1819 (2021)
30. Germain, H., Bourmaud, G., Lepetit, V.: S2DNet: learning accurate correspondences for sparse-to-dense feature matching. In: European Conference on Computer Vision (ECCV) (2020)
31. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: PREDATOR: registration of 3D point clouds with low overlap. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4267–4276 (2021)
32. Ke, T., Roumeliotis, S.I.: An efficient algebraic solution to the perspective-three-point problem. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
33. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: IEEE International Conference on Robotics and Automation (ICRA) (2016)
34. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
35. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DoF camera relocalization. In: IEEE International Conference on Computer Vision (ICCV) (2015)
36. Kneip, L., Scaramuzza, D., Siegwart, R.: A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
37. Laskar, Z., Melekhov, I., Kalia, S., Kannala, J.: Camera relocalization by computing pairwise relative poses using convolutional neural network. In: IEEE International Conference on Computer Vision (ICCV) Workshops (2017)
38. Li, X., Wang, S., Zhao, Y., Verbeek, J., Kannala, J.: Hierarchical scene coordinate classification and regression for visual localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11983–11992 (2020)
39. Li, Z., Snavely, N.: MegaDepth: learning single-view depth prediction from internet photos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
40. Liu, L., Campbell, D., Li, H., Zhou, D., Song, X., Yang, R.: Learning 2D-3D correspondences to solve the blind perspective-n-point problem (2020)
41. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
42. Luo, Z., et al.: ASLFeat: learning local features of accurate shape and localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6589–6598 (2020)

43. Mera-Trujillo, M., Smith, B., Fragoso, V.: Efficient scene compression for visual-based localization. In: 2020 International Conference on 3D Vision (3DV), pp. 1–10 (2020)
44. Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2666–2674 (2018)
45. Moreno-Noguer, F., Lepetit, V., Fua, P.: Pose priors for simultaneously solving alignment and correspondence. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5303, pp. 405–418. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88688-4_30
46. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. IEEE Trans. Pattern Anal. Mach. Intell. **36**(11), 2227–2240 (2014)
47. Ng, T., et al.: NinjaDesc: content-concealing visual descriptors via adversarial learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12797–12807 (2022)
48. Pittaluga, F., Koppal, S.J., Kang, S.B., Sinha, S.N.: Revealing scenes by inverting structure from motion reconstructions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 145–154 (2019)
49. Radwan, N., Valada, A., Burgard, W.: VLocNet++: deep multitask learning for semantic visual localization and odometry. IEEE Robot. Autom. Lett. **3**(4), 4407–4414 (2018)
50. Sarlin, P.-E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: robust hierarchical localization at large scale. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
51. Sarlin, P.-E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: learning feature matching with graph neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4938–4947 (2020)
52. Sarlin, P.-E., et al.: Back to the feature: learning robust camera localization from pixels to pose. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3247–3257 (2021)
53. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE Trans. Pattern Anal. Mach. Intell. **39**(9), 1744–1756 (2017)
54. Sattler, T., et al.: Benchmarking 6DoF outdoor visual localization in changing conditions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8601–8610 (2018)
55. Sattler, T., et al.: Are large-scale 3D models really necessary for accurate visual localization? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
56. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of CNN-based absolute camera pose regression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
57. Schönberger, J.L., Frahm, J.-M.: Structure-from-motion revisited. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
58. Schönberger, J.L., Pollefeys, M., Geiger, A., Sattler, T.: Semantic visual localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
59. Schönberger, J.L., Zheng, E., Frahm, J.-M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 501–518. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_31

60. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2733–2742 (2021)
61. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2930–2937 (2013)
62. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pac. J. Math. **21**(2), 343–348 (1967)
63. Speciale, P., Schonberger, J.L., Kang, S.B., Sinha, S.N., Pollefeys, M.: Privacy preserving image-based localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5493–5503 (2019)
64. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: detector-free local feature matching with transformers. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8922–8931 (2021)
65. Sun, W., Jiang, W., Trulls, E., Tagliasacchi, A., Yi, K.M.: ACNe: attentive context normalization for robust permutation-equivariant learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
66. Taira, H., et al.: InLoc: indoor visual localization with dense matching and view synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
67. Toft, C., et al.: Long-term visual localization revisited. IEEE Trans. Pattern Anal. Mach. Intell. **44**(4), 2074–2088 (2022)
68. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1808–1817 (2015)
69. Tran, N.-T., et al.: On-device scalable image-based localization via prioritized cascade search and fast one-many RANSAC. IEEE Trans. Image Process. **28**(4), 1675–1690 (2019)
70. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates Inc. (2017)
71. Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using LSTMs for structured feature correlation. In: IEEE International Conference on Computer Vision (ICCV) (2017)
72. Wang, Q., Zhou, X., Hariharan, B., Snavely, N.: Learning feature descriptors using camera pose supervision. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 757–774. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_44
73. Yang, L., Bai, Z., Tang, C., Li, H., Furukawa, Y., Tan, P.: SANet: scene agnostic network for camera localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 42–51 (2019)
74. Zhang, J., et al.: Learning two-view correspondences and geometry using order-aware network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
75. Zhou, Q., Sattler, T., Pollefeys, M., Leal-Taixe, L.: To learn or not to learn: visual localization from essential matrices. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3319–3326. IEEE (2020)

Part III

Conclusion and Outlook

8 Conclusion

In this thesis, we tackled visual localization, the task of estimating the camera pose of a query image w.r.t. a 3D scene. In most of the existing solutions, a scene is represented by a database of reference images with camera poses, except for the structure-based localization where one assumes an extra 3D model of the scene, which is commonly a point cloud reconstructed via SfM. Among different formulations of the solutions, we targeted relative pose-based localization and structure-based localization, which allows us to study methods with different types of representations. We then built upon their advantages and develop novel approaches to address the existing challenges in the existing formulations by leveraging modern deep learning techniques. We aimed to improve a localization system in terms of its scalability towards large-scale scenes, generalization across different types of scenes, and robustness and accuracy under various extreme environmental conditions.

In **chapter 5**, we proposed a generic framework for visual localization from essential matrices which is agnostic to how an essential matrix is computed. This characteristic enables in-depth study of various methods for computing essential matrices, ranging from purely hand-crafted to purely data-driven. Compared to structure-based localization which is the most accurate one but relies on an expensive scene point cloud, our framework is more flexible and light-weight as well as achieves competitive performance in accuracy and generalization when using the handcrafted SIFT feature and a 5-point solver to estimate essential matrices. With extensive evaluations and comparisons, we further show that different ways of utilizing deep learning for relative pose estimation led to highly different behaviors. Based on our experiments, learned feature extractor and matching do not suffer from generalization issue, while relative pose regression leads to the incapability of generalizing across indoor and outdoor scenes. We also found that the handcrafted SIFT features are still very competitive compared to our learned matches, which suggests the clear need for more advanced learning-based image matching techniques. Based on this work, we show the potential of RP-based localization for future scalable localization and point out clear working direction towards generalizable RP-based localization.

In **chapter 6**, we presented a new *detect-to-refine* paradigm for image matching that aims at pixel-level accurate correspondences. To tackle the challenging task of densely searching matches within a pair of images, we suggest splitting the task into stages, where the first stage focuses on capturing the semantic high-level information and the second stage focuses on the detailed structures inside local patches. To investigate the potential of this concept, we applied it to the existing correspondence network (NCNet) which is limited by the memory bottleneck to only match patch-level features. To *refine* match proposals *detected* by the pre-trained NCNet, we developed a novel refinement network to regress pixel-level matches from the local regions defined by those proposals and jointly predict confidence scores to reject outlier matches. Our network was weakly supervised by epipolar geometry to detect geometrically consistent correspondences without the need for ground-truth correspondences and can be trained end-to-end with the correspondence network. By evaluating our method on a variety of geometry tasks, we showed that our refinement network consistently and significantly improves the matching accuracy of the correspondence network baseline. Furthermore, our model trained to refine NCNet matches can be immediately leveraged to refine other match proposal networks without re-training. By directly applying our pre-trained refinement to the best fully-supervised matching method, we achieved state-of-the-art results on challenging long-term localization benchmark. To this end, we validated that *detect-to-refine* is a promising paradigm for future image matching that allows flexible and reasonable delegation of matching duties in a coarse-to-fine manner. While it has been widely explored how to refine match proposals by filtering the outliers, we are the first method that allows one to modify the locations of match proposals, leading to more advanced true refinement.

Finally, in **chapter 7**, we tested the potential of establishing 2D-3D correspondences via geometric-based matching for real-world visual localization that does not require visual descriptors for matching. We built upon the previous geometric-based matcher and significantly improve its robustness in handling keypoint outliers by leveraging the power of attention mechanisms to boost contextual information propagation. We further introduced a novel bearing vector representation of 3D points, which relieves the cross-modal challenge in matching 2D image keypoints to 3D point cloud with only geometric information. We showed that our method dramatically surpassed its prior work, making geometric-based matching feasible for realistic localization tasks. Plugging our geometric matcher into a hierarchical structure-based localization pipeline, we achieved performance on-par with the state-of-the-art absolute pose regressors. More importantly, our approach elegantly relieves structure-based localization from high storage demands, privacy concerns and the need for long-term descriptor maintenance. In conclusion, we verified that geometric-based matching is a promising option to pursue scalable, sustainable and reliable structure-based localization for large-scale scenes. Our work serves as a starting point along this new direction of geometric matching-based localization and we expect to inspire more research effort devoted to boosting its current performance in accuracy.

9 Future Work and Remarks

Extension to EssNet. In EssNet, we focused on studying the relative pose estimation module and fixed the image retrieval and pose estimation part. However, the quality of image retrieval has direct influence on both the following relative pose estimation and the triangulation step. Therefore, an interesting research direction arises from exploring the possibility of jointly learning image retrieval and relative pose together. In EssNet, we developed a classical RANSAC-based algorithm to estimate the absolute pose from multiple image pairs given their estimated relative poses. Another extension could be made by devising a learned pose triangulation algorithm as an alternative to this handcrafted process.

Extension to Patch2Pix. While Patch2Pix has the flexibility of modifying a match proposal by searching again within its local regions, it is still limited by the size of local regions. The local region has to be reasonable in size, otherwise the refinement becomes too expensive in computation and more challenging. The refinement is not possible if the errors of coarse matches are too larger. A direct extension to Patch2Pix is a joint proposal and refinement network learned in an end-to-end manner, where we want to reduce the number of proposals with big matching errors and leave the refinement network focusing on finding highly accurate matches.

Extension to GoMatch. In GoMatch, we assume keypoints are already given in a query image and are consistent with the 3D keypoints. However, we have shown that the performance of our method decreases largely with more than 50% of keypoint outliers. Besides pushing for more robust geometric-based matchers, another possibility is to reduce keypoint outliers from the beginning, where we can learn a keypoint outlier filtering function can be learned by exploring cues in keypoint distributions. Alternatively, one can also jointly learn the task of detection and geometric-based matching, where we expect keypoints to be directly tuned for the following matching step.

Correspondence networks. In correspondence networks, when an image is involved in two image pairs, most likely different sets of points will be matched to its corresponding paired image. This characteristic makes them less suitable for a structure-from-motion task as well as localizing an image against a pre-built point cloud. The naive solution to increase the keypoint repeatability, *i.e.*, a point being matching in multiple image pairs, is by applying quantization to matches such that the close-by keypoints involved in matches are merged into one. However such quantization leads to reduced matching accuracy, and thus is not optimal. As correspondence networks become popular and show promising matching performance compared to keypoint-based matching, it is a clear need to seek a better solution to the lack of repeatability issue. This can be addressed either in correspondence networks or in downstream tasks.

Relative pose-based localization is promising. One important conclusion we learned from EssNet [Zho+20] (chapter 5) is that relative pose-based localization can generalize if we perform matching-based instead of regression-based relative pose estimation. Our learned lesson is further confirmed by the very recent Mapfree [Arn+22], where they show that methods estimate relative pose estimated from 2D matches not only dominantly surpass those using relative pose regressions but also get closer to the state-of-the-art structure-based localization in accuracy and generalization. Compared to structure-based methods, relative pose-based localization has attracted much less attention in the past years since it is less accurate. However, relative pose-based localization has the significant advantage of not needing a 3D model, making it more scalable and suitable for city-scale localization. Therefore, we are very optimistic and excited about the future development in this direction.

Regression techniques: where to go? As we have introduced in previous sections, different regression techniques have been proposed to address localization including absolute pose regression (section 4.3), relative pose regression (section 4.5) and scene coordinate regression (section 4.4). While the original motivation is good, they are all faced with their own challenges.

Absolute pose regression (APR) was proposed to provide efficient and simple solution for localization requiring no scene map during inference. However, by formulation it is scene-dependent and its performance is highly related to the training data distribution [Sat+19] and relatively less accurate compared to other existing methods. It is very likely to fail when the testing data is distributed very differently from the training data. One way to relieve the generalization issue is by densely sampling training images from the scene, however, it makes the training longer, requires more storage to keep those training data and might not be feasible for large-scale scenes. Therefore, APR methods are more suitable for applications where one knows some priors about the testing scenario or where efficiency is more important than accuracy.

SCR learns an implicit 2D-3D matching function in a scene-dependent manner, which also does not need a 3D scene model during inference meanwhile achieves highly accurate performance for small-scale and indoor scenes. It is currently faced with the challenges of scaling to larger scenes and improving the performance in outdoor scenes. Solving those challenges will definitely bring SCR to a new level, making it a more competitive candidate for real-world applications. Different from APR, there is no direct constraint preventing a SCR to achieve those goals. There already exists some recent work showing progress in addressing these issues [Li+20a; TZ00]. Overall, we are highly interested in seeing future research about large-scale SCR.

While relative pose regression (RPR) by formulation is not limited to a specific scene, they suffer from limited generalization and accuracy compared to matching-based relative pose estimation [Zho+20; Arn+22]. Compared to APR and SCR methods, RPR has to rely on image retrieval and pose estimation (from relative poses) to localize a query image. Therefore, it is unclear what the advantages of using RPR is if it can not surpass matching-based solution.

BIBLIOGRAPHY

[ABD12] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. KAZE features. In: *European conference on computer vision*. Springer. 2012, pp. 214–227.

[ABI21] Yehya Abouelnaga, Mai Bui, and Slobodan Ilic. DistillPose: Lightweight Camera Localization Using Auxiliary Learning. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 7919–7924.

[Aga+11] Sameer Agarwal et al. Building rome in a day. In: *Communications of the ACM* 54.10 (2011), pp. 105–112.

[Aig+19] Dror Aiger, Haim Kaplan, Efi Kokiopoulou, Micha Sharir, and Bernhard Zeisl. General techniques for approximate incidences and their application to the camera posing problem. In: *arXiv preprint arXiv:1903.07047* (2019).

[Aig+21] Dror Aiger, Simon Lynen, Jan Hosang, and Bernhard Zeisl. Efficient Large Scale Inlier Voting for Geometric Vision Problems. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3243–3251.

[AOV12] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In: *2012 IEEE conference on computer vision and pattern recognition*. Ieee. 2012, pp. 510–517.

[Ara+16] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5297–5307.

[Arn+22] Eduardo Arnold et al. Map-Free Visual Relocalization: Metric Pose Relative to a Single Image. In: *European Conference on Computer Vision*. Springer. 2022, pp. 690–708.

[Art+09] Clemens Arth, Daniel Wagner, Manfred Klopschitz, Arnold Irschara, and Dieter Schmalstieg. Wide area localization on mobile phones. In: *2009 8th ieee international symposium on mixed and augmented reality*. IEEE. 2009, pp. 73–82.

[Art+11] Clemens Arth, Manfred Klopschitz, Gerhard Reitmayr, and Dieter Schmalstieg. Real-time self-localization from panoramic images on mobile devices. In: *2011 10th ieee international symposium on mixed and augmented reality*. IEEE. 2011, pp. 37–46.

[Avr+10] Yannis Avrithis, Yannis Kalantidis, Giorgos Tolias, and Evaggelos Spyrou. Retrieving landmark and non-landmark images from community photo collections. In: *Proceedings of the 18th ACM international conference on Multimedia*. 2010, pp. 153–162.

[AZ12] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 2911–2918.

[Azi+15] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 36–45.

[Bab+14] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In: *European conference on computer vision*. Springer. 2014, pp. 584–599.

[Bal+16] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In: *Bmvc*. Vol. 1. 2. 2016, p. 3.

[Bal+17] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5173–5182.

[Bar+19] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 5836–5844.

[Bar+22] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5470–5479.

[Bau00] Adam Baumberg. Reliable feature matching across widely separated views. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*. Vol. 1. IEEE. 2000, pp. 774–781.

[BL15] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. In: *arXiv preprint arXiv:1510.07493* (2015).

[Bla+20] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 38–39.

[BLP18] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In: *European Conference on Computer Vision*. 2018, pp. 751–767.

[BM22] Fabio Bellavia and Dmytro Mishkin. HarrisZ+: Harris corner selection for next-gen image matching pipelines. In: *Pattern Recognition Letters* 158 (2022), pp. 141–147.

[BMC22] 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*. 2022, pp. 4878–4888.

[BR18] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4654–4662.

[BR19a] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7525–7534.

[BR19b] Eric Brachmann and Carsten Rother. Neural-guided RANSAC: Learning where to sample model hypotheses. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4322–4331.

[BR21] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. In: *IEEE transactions on pattern analysis and machine intelligence* 44.9 (2021), pp. 5847–5865.

[Bra+16] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3364–3372.

[Bra+17] Eric Brachmann et al. Dsac-differentiable ransac for camera localization. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6684–6692.

[Bra+18] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2616–2625.

[Bra+21] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6218–6228.

[BTG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In: *European conference on computer vision*. Springer. 2006, pp. 404–417.

[BTV11] F Bellavia, D Tegolo, and C Valenti. Improving Harris corner selection strategy. In: *IET Computer Vision* 5.2 (2011), pp. 87–96.

[Cal+10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In: *European conference on computer vision*. Springer. 2010, pp. 778–792.

[Cam+19] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7653–7662.

[Cao+10] Yang Cao, Changhu Wang, Zhiwei Li, Liqing Zhang, and Lei Zhang. Spatial-bag-of-features. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 3352–3359.

[CAS20] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In: *European Conference on Computer Vision*. Springer. 2020, pp. 726–743.

[Cav+17] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Luigi Di Stefano, and Philip HS Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4457–4466.

[Cav+19a] Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In: *2019 International Conference on 3D Vision (3DV)*. IEEE. 2019, pp. 564–573.

[Cav+19b] Tommaso Cavallari et al. Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. In: *IEEE transactions on pattern analysis and machine intelligence* 42.10 (2019), pp. 2465–2477.

[CF80] Ning-San Chang and King-sun Fu. A relational database system for images. In: *Pictorial Information Systems* (1980), pp. 288–321.

[CH18] Gabriela Csurka and Martin Humenberger. From handcrafted to deep local invariant features. In: *arXiv preprint arXiv:1807.10254* 2 (2018), p. 1.

[Cha+22] Ming-Fang Chang, Yipu Zhao, Rajvi Shah, Jakob J Engel, Michael Kaess, and Simon Lucey. Long-term Visual Map Sparsification with Heterogeneous GNN. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2406–2415.

[Che+13] Jie Chen, Vili Kellokumpu, Guoying Zhao, and Matti Pietikäinen. RLBP: Robust Local Binary Pattern. In: *BMVC*. 2013.

[Che+19] Wentao Cheng, Weisi Lin, Kan Chen, and Xinfeng Zhang. Cascaded parallel filtering for memory-efficient image-based localization. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1032–1041.

[Che+21] Hongkai Chen et al. Learning to match features with seeded graph matching network. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6301–6310.

[Che+22a] Hongkai Chen et al. ASpanFormer: Detector-Free Image Matching with Adaptive Span Transformer. In: *European Conference on Computer Vision*. Springer. 2022, pp. 20–36.

[Che+22b] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Adrian Prisacariu. DFNet: Enhance Absolute Pose Regression with Direct Feature Matching. In: *arXiv preprint arXiv:2204.00559* (2022).

[CHL05] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).* Vol. 1. IEEE. 2005, pp. 539–546.

[Chu+07] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In: *2007 IEEE 11th International Conference on Computer Vision.* IEEE. 2007, pp. 1–8.

[CKM08] Robert Castle, Georg Klein, and David W Murray. Video-rate localization in multiple maps for wearable augmented reality. In: *2008 12th IEEE International Symposium on Wearable Computers.* IEEE. 2008, pp. 15–22.

[CKS21] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021, pp. 15668–15678.

[Cla+17] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2017, pp. 6856–6864.

[CLG20] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In: *European Conference on Computer Vision.* Springer. 2020, pp. 244–261.

[CMK03] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In: *Joint Pattern Recognition Symposium.* Springer. 2003, pp. 236–243.

[CN12] Siddharth Choudhary and PJ Narayanan. Visibility probability structure from sfm datasets and applications. In: *European conference on computer vision.* Springer. 2012, pp. 130–143.

[CPM09] Ondrej Chum, Michal Perd'och, and Jiri Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE. 2009, pp. 17–24.

[CR97] Wan-Ching Chen and Peter Rockett. Bayesian labelling of corners using a grey-level corner image model. In: *Proceedings of International Conference on Image Processing.* Vol. 1. IEEE. 1997, pp. 687–690.

[CS13] Song Cao and Noah Snavely. Graph-based discriminative learning for location recognition. In: *Proceedings of the ieee conference on computer vision and pattern recognition.* 2013, pp. 700–707.

[CS14] Song Cao and Noah Snavely. Minimal scene descriptions from structure from motion models. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2014, pp. 461–468.

[Csu+04] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 1-22. Prague. 2004, pp. 1–2.

[Cut13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In: *Advances in neural information processing systems* 26 (2013).

[CWM05] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 772–779.

[CWP21] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute pose regression with photometric consistency. In: *2021 International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 1175–1185.

[CYT19] Zhi Chen, Fan Yang, and Wenbing Tao. Gla-net: An attention network with guided loss for mismatch removal. In: *arXiv preprint arXiv:1909.13092* (2019).

[Dai+17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5828–5839.

[dAn+13] Emmanuel d'Angelo, Laurent Jacques, Alexandre Alahi, and Pierre Vandergheynst. From bits to images: Inversion of local binary descriptors. In: *IEEE transactions on pattern analysis and machine intelligence* 36.5 (2013), pp. 874–887.

[Dav+07] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. In: *IEEE transactions on pattern analysis and machine intelligence* 29.6 (2007), pp. 1052–1067.

[DB16] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4829–4837.

[Den+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[Den+19] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4690–4699.

[Di +18] Paolo Di Febbo, Carlo Dal Mutto, Kinh Tieu, and Stefano Mattoccia. Kcnn: Extremely-efficient hardware keypoint detection with a compact convolutional neural network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 682–690.

[DKS95] PGT Dias, Ashraf A Kassim, and V Srinivasan. A neural network based corner detection method. In: *Proceedings of ICNN'95-International Conference on Neural Networks*. Vol. 4. IEEE. 1995, pp. 2116–2120.

[DMR16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. In: *arXiv preprint arXiv:1606.03798* (2016).

[DMR17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. In: *arXiv preprint arXiv:1707.07410* (2017).

[DMR18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 224–236.

[Don+09] Zilong Dong, Guofeng Zhang, Jiaya Jia, and Hujun Bao. Keyframe-based real-time camera tracking. In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 1538–1545.

[DS14] Michael Donoser and Dieter Schmalstieg. Discriminative feature-to-point matching in image-based localization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 516–523.

[DS15] Jingming Dong and Stefano Soatto. Domain-size pooling in local descriptors: DSP-SIFT. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 5097–5106.

[Dus+19] Mihai Dusmanu et al. D2-net: A trainable cnn for joint description and detection of local features. In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2019, pp. 8092–8101.

[Dus+21a] Mihai Dusmanu, Ondrej Miksik, Johannes L Schönberger, and Marc Pollefeys. Cross-descriptor visual localization and mapping. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6058–6067.

[Dus+21b] Mihai Dusmanu, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy-preserving image features via adversarial affine subspace embeddings. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14267–14277.

[Ebe+19] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 253–262.

[ELJ18] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018, pp. 0–0.

[FB81] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[FM90] Olivier D Faugeras and Steve Maybank. Motion from point matches: multiplicity of solutions. In: *International Journal of Computer Vision* 4.3 (1990), pp. 225–246.

[För94] Wolfgang Förstner. A framework for low level feature extraction. In: *European Conference on Computer Vision*. Springer. 1994, pp. 383–394.

[Fra+10] Jan-Michael Frahm et al. Building rome on a cloudless day. In: *European conference on computer vision*. Springer. 2010, pp. 368–381.

[Gao+03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. In: *IEEE transactions on pattern analysis and machine intelligence* 25.8 (2003), pp. 930–943.

[GBL19] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. Sparse-to-dense hypercolumn matching for long-term visual localization. In: *2019 International Conference on 3D Vision (3DV)*. IEEE. 2019, pp. 513–523.

[GBL20] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2Dnet: learning image features for accurate sparse-to-dense matching. In: *European Conference on Computer Vision*. Springer. 2020, pp. 626–643.

[Gep+19] Marcel Geppert, Peidong Liu, Zhaopeng Cui, Marc Pollefeys, and Torsten Sattler. Efficient 2d-3d matching for multi-camera visual localization. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5972–5978.

[Gep+20] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schönberger, and Marc Pollefeys. Privacy preserving structure-from-motion. In: *European Conference on Computer Vision*. Springer. 2020, pp. 333–350.

[Gep+21] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schonberger, and Marc Pollefeys. Privacy preserving localization and mapping from uncalibrated cameras. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1809–1819.

[GHT11] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. In: *International journal of computer vision* 94.3 (2011), pp. 335–360.

[GKS13] Tiezheng Ge, Qifa Ke, and Jian Sun. Sparse-Coded Features for Image Retrieval. In: *BMVC*. 2013, pp. 132–1.

[GL06] Iryna Gordon and David G Lowe. What and where: 3D object recognition with accurate pose. In: *Toward category-level object recognition*. Springer, 2006, pp. 67–82.

[GLB21] Hugo Germain, Vincent Lepetit, and Guillaume Bourmaud. Visual Correspondence Hallucination. In: *International Conference on Learning Representations*. 2021.

[Gon+14] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In: *European conference on computer vision*. Springer. 2014, pp. 392–407.

[Gor+16] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In: *European conference on computer vision*. Springer. 2016, pp. 241–257.

[Gor+17] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. In: *International Journal of Computer Vision* 124.2 (2017), pp. 237–254.

[GPM10] Raj Gupta, Harshal Patil, and Anurag Mittal. Robust order-based methods for feature description. In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 334–341.

[Guz+14] Abner Guzman-Rivera et al. Multi-output learning for camera relocalization. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1114–1121.

[HA15] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In: *International workshop on similarity-based pattern recognition*. Springer. 2015, pp. 84–92.

[Hal+06] Alaa Halawani, Alexandra Teynor, Lokesh Setia, Gerd Brunner, and Hans Burkhardt. Fundamentals and Applications of Image Retrieval: An Overview. In: *Datenbank-Spektrum* 18.14-23 (2006), p. 6.

[Han+15] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3279–3286.

[Hau+21] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14141–14152.

[HE08] James Hays and Alexei A Efros. IM2GPS: estimating geographic information from a single image. In: *2008 ieee conference on computer vision and pattern recognition*. IEEE. 2008, pp. 1–8.

[Hen+19] Lionel Heng et al. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 4695–4702.

[HL12] Richard Hartley and Hongdong Li. An efficient hidden variable approach to minimal-case camera motion estimation. In: *IEEE transactions on pattern analysis and machine intelligence* 34.12 (2012), pp. 2303–2314.

[HLS18] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 596–605.

[HPS09] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with local binary patterns. In: *Pattern recognition* 42.3 (2009), pp. 425–436.

[HS+88] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In: *Alvey vision conference.* Vol. 15. 50. Manchester, UK. 1988, pp. 10–5244.

[Hua+22] Dihe Huang et al. Adaptive Assignment for Geometry Aware Local Feature Matching. In: *arXiv preprint arXiv:2207.08427* (2022).

[Hum+22] Martin Humenberger et al. Investigating the Role of Image Retrieval for Visual Localization. In: *International Journal of Computer Vision* (2022), pp. 1–26.

[HZ03] Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.

[Irs+09] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE. 2009, pp. 2599–2606.

[JDS08] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In: *European conference on computer vision.* Springer. 2008, pp. 304–317.

[JDS09] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In: *2009 IEEE conference on computer vision and pattern recognition.* IEEE. 2009, pp. 1169–1176.

[Jég+10] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In: *2010 IEEE computer society conference on computer vision and pattern recognition.* IEEE. 2010, pp. 3304–3311.

[Jég+11] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. In: *IEEE transactions on pattern analysis and machine intelligence* 34.9 (2011), pp. 1704–1716.

[Jia+21] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2021, pp. 6207–6217.

[Jin+21] Yuhe Jin et al. Image matching across wide baselines: From paper to practice. In: *International Journal of Computer Vision* 129.2 (2021), pp. 517–547.

[JZ14] Hervé Jégou and Andrew Zisserman. Triangulation embedding and democratic aggregation for image search. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014, pp. 3310–3317.

[KC16] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In: *2016 IEEE international conference on Robotics and Automation (ICRA).* IEEE. 2016, pp. 4762–4769.

[KC17] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 5974–5983.

[KCR16] Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5385–5394.

[KGC15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2938–2946.

[KH14] Hiroharu Kato and Tatsuya Harada. Image reconstruction from bag-of-visual-words. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 955–962.

[KM15] Kishore Reddy Konda and Roland Memisevic. Learning visual odometry with a convolutional network. In: *VISAPP (1)*. 2015, pp. 486–490.

[KMO16] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In: *European conference on computer vision*. Springer. 2016, pp. 685–701.

[Kol+22] Manuel Kolmet, Qunjie Zhou, Aljoša Ošep, and Laura Leal-Taixe. Text2Pos: Text-to-Point-Cloud Cross-Modal Localization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 6687–6696.

[KR17] Tong Ke and Stergios I Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7225–7233.

[KSP10] Jan Knopp, Josef Sivic, and Tomas Pajdla. Avoiding confusing features in place recognition. In: *European Conference on Computer Vision*. Springer. 2010, pp. 748–761.

[KSS11] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: *CVPR 2011*. IEEE. 2011, pp. 2969–2976.

[Kuk+17] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla. A clever elimination strategy for efficient minimal solvers. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4912–4921.

[LaM+05] Anthony LaMarca et al. Place lab: Device positioning using radio beacons in the wild. In: *International conference on pervasive computing*. Springer. 2005, pp. 116–133.

[Lar+19] Viktor Larsson, Torsten Sattler, Zuzana Kukelova, and Marc Pollefeys. Revisiting radial distortion absolute pose. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1062–1071.

[Las+17] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 929–938.

[LCS11] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. BRISK: Binary robust invariant scalable keypoints. In: *2011 International conference on computer vision*. Ieee. 2011, pp. 2548–2555.

[Lee+22] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. Correlation Verification for Image Retrieval. In: 2022.

[LF06] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. In: *IEEE transactions on pattern analysis and machine intelligence* 28.9 (2006), pp. 1465–1479.

[LH06] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 1. IEEE. 2006, pp. 630–633.

[Li+12] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In: *European conference on computer vision*. Springer. 2012, pp. 15–29.

[Li+20a] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11983–11992.

[Li+20b] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17346–17357.

[Li+22] Kunhong Li, Longguang Wang, Li Liu, Qing Ran, Kai Xu, and Yulan Guo. Decoupling Makes Weakly Supervised Local Feature Better. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 15838–15848.

[Lim+12] Hyon Lim, Sudipta N Sinha, Michael F Cohen, and Matthew Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 1043–1050.

[Liu+19] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns. In: *Advances in Neural Information Processing Systems* 32 (2019).

[Liu+20] Liu Liu, Dylan Campbell, Hongdong Li, Dingfu Zhou, Xibin Song, and Ruigang Yang. Learning 2d-3d correspondences to solve the blind perspective-n-point problem. In: *arXiv preprint arXiv:2003.06752* (2020).

[Liu+21] Yuan Liu, Lingjie Liu, Cheng Lin, Zhen Dong, and Wenping Wang. Learnable motion coherence for correspondence pruning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3237–3246.

[LLD17] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2372–2381.

[LM22] Axel Barroso Laguna and Krystian Mikolajczyk. Key. Net: Keypoint Detection by Handcrafted and Learned CNN Filters Revisited. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[Low99] David G Lowe. Object recognition from local scale-invariant features. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[LSH10] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In: *European Conference on Computer Vision*. Springer. 2010, pp. 791–804.

[Luo+18] Zixin Luo et al. Geodesc: Learning local descriptors by integrating geometry constraints. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 168–183.

[Luo+19] Zixin Luo et al. Contextdesc: Local descriptor augmentation with cross-modality context. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2527–2536.

[Luo+20] Zixin Luo et al. Aslfeat: Learning local features of accurate shape and localization. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6589–6598.

[LV16] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In: *European conference on computer vision*. Springer. 2016, pp. 100–117.

[Lyn+15] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In: *Robotics: Science and Systems*. Vol. 1. 2015, p. 1.

[LZP11] Steven Lanzisera, David Zats, and Kristofer SJ Pister. Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization. In: *IEEE Sensors Journal* 11.3 (2011), pp. 837–845.

[Ma+21] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. In: *International Journal of Computer Vision* 129.1 (2021), pp. 23–79.

[Mao+22] Runyu Mao, Chen Bai, Yatong An, Fengqing Zhu, and Cheng Lu. 3DG-STFM: 3D Geometric Guided Student-Teacher Feature Matching. In: *European Conference on Computer Vision*. Springer. 2022, pp. 125–142.

[Mar+21] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7210–7219.

[Mat+04] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. In: *Image and vision computing* 22.10 (2004), pp. 761–767.

[MC21] Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. In: *IEEE Access* 9 (2021), pp. 19516–19547.

[Mel+17a] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In: *Proceedings of the IEEE international conference on computer vision workshops*. 2017, pp. 879–886.

[Mel+17b] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer. 2017, pp. 675–687.

[Mid+14] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In: *European conference on computer vision*. Springer. 2014, pp. 268–283.

[Mik+13] Andrej Mikulik, Michal Perdoch, Ondrej Chum, and Jiri Matas. Learning vocabularies over a fine quantization. In: *International journal of computer vision* 103.1 (2013), pp. 163–175.

[Mil+21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[Mis+17] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In: *Advances in neural information processing systems* 30 (2017).

[ML14] Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. In: *IEEE transactions on pattern analysis and machine intelligence* 36.11 (2014), pp. 2227–2240.

[MMT15] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. In: *IEEE transactions on robotics* 31.5 (2015), pp. 1147–1163.

[Mor+22] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by NeRF synthesis. In: *Conference on Robot Learning*. PMLR. 2022, pp. 1347–1356.

[Mor77] Hans P Moravec. Techniques towards automatic visual obstacle avoidance. In: (1977).

[MS01] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.* Vol. 1. IEEE. 2001, pp. 525–531.

[MS02] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In: *European conference on computer vision.* Springer. 2002, pp. 128–142.

[MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. In: *International journal of computer vision* 60.1 (2004), pp. 63–86.

[MS05] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In: *IEEE transactions on pattern analysis and machine intelligence* 27.10 (2005), pp. 1615–1630.

[MSF20] Marcela Mera-Trujillo, Benjamin Smith, and Victor Fragoso. Efficient scene compression for visual-based localization. In: *2020 International Conference on 3D Vision (3DV).* IEEE. 2020, pp. 1–10.

[MT17] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. In: *IEEE transactions on robotics* 33.5 (2017), pp. 1255–1262.

[MV15] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 5188–5196.

[MY09] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. In: *SIAM journal on imaging sciences* 2.2 (2009), pp. 438–469.

[NB17] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2017, pp. 1525–1530.

[New+11] Richard A Newcombe et al. Kinectfusion: Real-time dense surface mapping and tracking. In: *2011 10th IEEE international symposium on mixed and augmented reality.* Ieee. 2011, pp. 127–136.

[Ng+22] Tony Ng et al. NinjaDesc: Content-Concealing Visual Descriptors via Adversarial Learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2022, pp. 12797–12807.

[Nis04] David Nistér. An efficient solution to the five-point relative pose problem. In: *IEEE transactions on pattern analysis and machine intelligence* 26.6 (2004), pp. 756–770.

[Noh+17] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 3456–3465.

[NS06] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. Ieee. 2006, pp. 2161–2168.

[Ono+18] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In: *Advances in neural information processing systems* 31 (2018).

[OPM02] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. In: *IEEE Transactions on pattern analysis and machine intelligence* 24.7 (2002), pp. 971–987.

[Ozu+09] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. In: *IEEE transactions on pattern analysis and machine intelligence* 32.3 (2009), pp. 448–461.

[PC+19] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[PCM09] Michal Perd'och, Ondrej Chum, and Jiri Matas. Efficient representation of local geometry for large scale object retrieval. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 9–16.

[PD07] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In: *2007 IEEE conference on computer vision and pattern recognition*. IEEE. 2007, pp. 1–8.

[Per+10] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 3384–3391.

[Phi+07] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In: *2007 IEEE conference on computer vision and pattern recognition*. IEEE. 2007, pp. 1–8.

[Pit+19] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 145–154.

[PKS22] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. MeshLoc: Mesh-Based Visual Localization. In: *European Conference on Computer Vision*. Springer. 2022, pp. 589–609.

[Pra+22] Shraman Pramanick, Ewa M Nowara, Joshua Gleason, Carlos D Castillo, and Rama Chellappa. Where in the World is this Image? Transformer-based Geo-localization in the Wild. In: *European Conference on Computer Vision.* Springer. 2022, pp. 196–215.

[Qin+11] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In: *CVPR 2011.* IEEE. 2011, pp. 777–784.

[RAS17] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 6148–6157.

[RAS20] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In: *European conference on computer vision.* Springer. 2020, pp. 605–621.

[RC04] Duncan P Robertson and Roberto Cipolla. An Image-Based System for Urban Navigation. In: *Bmvc.* Vol. 19. 51. Citeseer. 2004, p. 165.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In: *European conference on computer vision.* Springer. 2006, pp. 430–443.

[Ren+15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems* 28 (2015).

[Rev+19a] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2019, pp. 5107–5116.

[Rev+19b] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In: *Advances in neural information processing systems* 32 (2019).

[RFS88] Nick Roussopoulos, Christos Faloutsos, and Timos Sellis. An efficient pictorial database system for PSQL. In: *IEEE transactions on software engineering* 14.5 (1988), pp. 639–650.

[RO13] Andrew Richardson and Edwin Olson. Learning convolutional filters for interest point detection. In: *2013 IEEE International Conference on Robotics and Automation.* IEEE. 2013, pp. 631–637.

[Roc+18] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In: *Advances in neural information processing systems* 31 (2018).

[RPD08] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. In: *IEEE transactions on pattern analysis and machine intelligence* 32.1 (2008), pp. 105–119.

[RTC18] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. In: *IEEE transactions on pattern analysis and machine intelligence* 41.7 (2018), pp. 1655–1668.

[Rub+11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In: *2011 International conference on computer vision*. Ieee. 2011, pp. 2564–2571.

[RVB18] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4407–4414.

[SAC19] Oriane Siméoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11651–11660.

[Sar+18] Paul-Edouard Sarlin, Frédéric Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In: *Conference on Robot Learning*. PMLR. 2018, pp. 456–465.

[Sar+19] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12716–12725.

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

[Sat+12] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In: *BMVC*. Vol. 1. 2. 2012, p. 4.

[Sat+15] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2102–2110.

[Sat+18] Torsten Sattler et al. Benchmarking 6dof outdoor visual localization in changing conditions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8601–8610.

[Sat+19] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3302–3312.

[Sav+17] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1822–1830.

[SB97] Stephen M Smith and J Michael Brady. SUSAN—a new approach to low level image processing. In: *International journal of computer vision* 23.1 (1997), pp. 45–78.

[SBS07] Grant Schindler, Matthew Brown, and Richard Szeliski. City-scale location recognition. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–7.

[Sch+16] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In: *European conference on computer vision*. Springer. 2016, pp. 501–518.

[SF16] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4104–4113.

[SFK21] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2733–2742.

[SH+04] Chanop Silpa-Anan, Richard Hartley, et al. Localization using an imagemap. In: *Australasian Conf. on Robotics and Automation*. Vol. 162. 2004.

[She+13] Xiaohui Shen, Zhe Lin, Jonathan Brandt, and Ying Wu. Spatially-constrained similarity measurefor large-scale object retrieval. In: *IEEE transactions on pattern analysis and machine intelligence* 36.6 (2013), pp. 1229–1241.

[Shi+22] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. ClusterGNN: Cluster-based Coarse-to-Fine Graph Neural Network for Efficient Feature Matching. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12517–12526.

[Shi+94] Jianbo Shi et al. Good features to track. In: *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE. 1994, pp. 593–600.

[Sho+13] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2930–2937.

[Sim+15] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 118–126.

[SK67] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348.

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 815–823.

[SL04] Iryna Skrypnyk and David G Lowe. Scene modelling, recognition and tracking with invariant image features. In: *Third IEEE and ACM international symposium on mixed and augmented reality.* IEEE. 2004, pp. 110–119.

[SLK11] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In: *2011 International Conference on Computer Vision.* IEEE. 2011, pp. 667–674.

[SLK16] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. In: *IEEE transactions on pattern analysis and machine intelligence* 39.9 (2016), pp. 1744–1756.

[SLL01] Stephen Se, David Lowe, and Jim Little. Vision-based mobile robot localization and mapping using scale-invariant features. In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164).* Vol. 2. IEEE. 2001, pp. 2051–2058.

[SLL02] Stephen Se, David Lowe, and Jim Little. Global localization using distinctive visual features. In: *IEEE/RSJ international conference on intelligent robots and systems.* Vol. 1. IEEE. 2002, pp. 226–231.

[SLM14] Yuxiang Sun, Ming Liu, and Max Q-H Meng. WiFi signal strength-based robot indoor localization. In: *2014 IEEE International Conference on Information and Automation (ICIA).* IEEE. 2014, pp. 250–256.

[SM96] Cordelia Schmid and Roger Mohr. Combining greyvalue invariants with local constraints for object recognition. In: *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE. 1996, pp. 872–877.

[Son+17] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 1746–1754.

[Soo+13] Hyun Soo Park, Yu Wang, Eriko Nurvitadhi, James C Hoe, Yaser Sheikh, and Mei Chen. 3d point cloud reduction using mixed-integer quadratic programming. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.* 2013, pp. 229–236.

[SQ17] Ehab Salahat and Murad Qasaimeh. Recent advances in features extraction and description algorithms: A comprehensive survey. In: *2017 IEEE international conference on industrial technology (ICIT).* IEEE. 2017, pp. 1059–1063.

[SSS06] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. In: *ACM siggraph 2006 papers.* 2006, pp. 835–846.

[Ste+08] Henrik Stewénius, David Nistér, Fredrik Kahl, and Frederik Schaffalitzky. A minimal solution for relative pose with unknown focal length. In: *Image and Vision Computing* 26.7 (2008), pp. 871–877.

[Sun+20] Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. Acne: Attentive context normalization for robust permutation-equivariant learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2020, pp. 11286–11295.

[Sun+21] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2021, pp. 8922–8931.

[Sva+14] Linus Svarm, Olof Enqvist, Magnus Oskarsson, and Fredrik Kahl. Accurate localization and pose estimation for large 3d models. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2014, pp. 532–539.

[Svä+16] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. In: *IEEE transactions on pattern analysis and machine intelligence* 39.7 (2016), pp. 1455–1461.

[SVJ18] Soham Saha, Girish Varma, and CV Jawahar. Improved visual relocalization by discovering anchor points. In: *arXiv preprint arXiv:1811.04370* (2018).

[SZ03] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In: *Computer Vision, IEEE International Conference on.* Vol. 3. IEEE Computer Society. 2003, pp. 1470–1470.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In: *arXiv preprint arXiv:1409.1556* (2014).

[Sze+15] Christian Szegedy et al. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 1–9.

[Sze22] Richard Szeliski. Computer vision: algorithms and applications. Springer Nature, 2022.

[SZF20] Hui Sun, Wenju Zhou, and Minrui Fei. A survey on graph matching in computer vision. In: *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI).* IEEE. 2020, pp. 225–230.

[Tai+18] Hajime Taira et al. InLoc: Indoor visual localization with dense matching and view synthesis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018, pp. 7199–7209.

[Tan+09] Feng Tang, Suk Hwan Lim, Nelson L Chang, and Hai Tao. A novel feature descriptor invariant to complex brightness changes. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE. 2009, pp. 2631–2638.

[Tan+22] Dongli Tan et al. ECO-TR: Efficient Correspondences Finding via Coarse-to-Fine Refinement. In: *European Conference on Computer Vision.* Springer. 2022, pp. 317–334.

[Tei+19] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2019, pp. 5109–5118.

[TFT20] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14254–14265.

[TFW17] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 661–669.

[TH98] Miroslav Trajković and Mark Hedley. Fast corner detection. In: *Image and vision computing* 16.2 (1998), pp. 75–87.

[Tia+19] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2019, pp. 11016–11025.

[Tia+20] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2d: Keypoint extraction with describe to detect approach. In: *Proceedings of the Asian Conference on Computer Vision.* 2020.

[TJ14] Giorgos Tolias and Hervé Jégou. Visual query expansion with or without geometry: refining local descriptors by feature aggregation. In: *Pattern recognition* 47.10 (2014), pp. 3466–3476.

[TLF09] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. In: *IEEE transactions on pattern analysis and machine intelligence* 32.5 (2009), pp. 815–830.

[TM+08] Tinne Tuytelaars, Krystian Mikolajczyk, et al. Local invariant feature detectors: a survey. In: *Foundations and trends® in computer graphics and vision* 3.3 (2008), pp. 177–280.

[Tof+20] Carl Toft et al. Long-term visual localization revisited. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[Tok+21] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixé. Coming down to earth: Satellite-to-street view synthesis for geo-localization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021, pp. 6488–6497.

[Tor+15] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 1808–1817.

[Tri04] Bill Triggs. Detecting keypoints with stable position, orientation, and scale under illumination changes. In: *European conference on computer vision.* Springer. 2004, pp. 100–113.

[TSJ15] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In: *arXiv preprint arXiv:1511.05879* (2015).

[TW09] Matthew Toews and William Wells. Sift-rank: Ordinal description for invariant feature correspondence. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE. 2009, pp. 172–177.

[TYO21] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. Instance-level image retrieval using reranking transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2021, pp. 12105–12115.

[TZ00] Philip HS Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. In: *Computer vision and image understanding* 78.1 (2000), pp. 138–156.

[Val+15] Julien Valentin, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip HS Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 4400–4408.

[Val+16] Julien Valentin et al. Learning to navigate the energy landscape. In: *2016 Fourth International Conference on 3D Vision (3DV).* IEEE. 2016, pp. 323–332.

[Vas+17] Ashish Vaswani et al. Attention is all you need. In: *Advances in neural information processing systems* 30 (2017).

[Ven+14] Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg. Global localization from monocular slam on a mobile phone. In: *IEEE transactions on visualization and computer graphics* 20.4 (2014), pp. 531–539.

[Ver+15] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 5279–5288.

[Von+13] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In: *Proceedings of the IEEE International Conference on Computer Vision.* 2013, pp. 1–8.

[VRB18] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In: *2018 IEEE international conference on robotics and automation (ICRA).* IEEE. 2018, pp. 6939–6946.

[Wal+17] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In: *Proceedings of the IEEE International Conference on Computer Vision.* 2017, pp. 627–637.

[Wan+15] Zhenhua Wang, Bin Fan, Gang Wang, and Fuchao Wu. Exploring local and overall ordinal information for robust feature description. In: *IEEE transactions on pattern analysis and machine intelligence* 38.11 (2015), pp. 2198–2211.

[Wan+20a] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 06. 2020, pp. 10393–10401.

[Wan+20b] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In: *European Conference on Computer Vision*. Springer. 2020, pp. 757–774.

[Wan+22] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zhen. TransVPR: Transformer-based place recognition with multi-level attention aggregation. In: *arXiv preprint arXiv:2201.02001* (2022).

[WDT21] Dominik Winkelbauer, Maximilian Denninger, and Rudolph Triebel. Learning to localize in new environments from synthetic training data. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 5840–5846.

[Wei+18] Xing Wei, Yue Zhang, Yihong Gong, and Nanning Zheng. Kernelized subspace pooling for deep local descriptors. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1867–1875.

[WEZ21] Olivia Wiles, Sebastien Ehrhardt, and Andrew Zisserman. Co-attention for conditioned image matching. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15920–15929.

[WF17] Erik Wijmans and Yasutaka Furukawa. Exploiting 2d floorplan for building-scale panorama rgbd alignment. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 308–316.

[WFW11] Zhenhua Wang, Bin Fan, and Fuchao Wu. Local intensity order pattern for feature description. In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 603–610.

[WIB11] Andreas Wendel, Arnold Irschara, and Horst Bischof. Natural landmark-based monocular localization for MAVs. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 5792–5799.

[WKP16] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In: *European Conference on Computer Vision*. Springer. 2016, pp. 37–55.

[Wu+11] Changchang Wu et al. VisualSFM: A visual structure from motion system. In: (2011).

[Wu13] Changchang Wu. Towards linear-time incremental structure from motion. In: *2013 International Conference on 3D Vision-3DV 2013*. IEEE. 2013, pp. 127–134.

[Xue+20] Fei Xue, Xin Wu, Shaojun Cai, and Junqiu Wang. Learning multi-view camera relocalization with graph neural networks. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 11372–11381.

[Yan+16] Junchi Yan, Xu-Cheng Yin, Weiyao Lin, Cheng Deng, Hongyuan Zha, and Xiaokang Yang. A short survey of recent advances in graph matching. In: *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval.* 2016, pp. 167–174.

[Yan+19] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. Sanet: Scene agnostic network for camera localization. In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2019, pp. 42–51.

[Yen+21] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 1323–1330.

[Yi+16] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In: *European conference on computer vision.* Springer. 2016, pp. 467–483.

[Yi+18] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2018, pp. 2666–2674.

[Zha+17] Xu Zhang, Felix X Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In: *Proceedings of the IEEE international conference on computer vision.* 2017, pp. 4595–4603.

[Zha+19a] Jiahui Zhang et al. Learning two-view correspondences and geometry using order-aware network. In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2019, pp. 5845–5854.

[Zha+19b] Chen Zhao, Zhiguo Cao, Chi Li, Xin Li, and Jiaqi Yang. Nm-net: Mining reliable neighbors for robust feature correspondences. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2019, pp. 215–224.

[Zho+10] Wengang Zhou, Yijuan Lu, Houqiang Li, Yibing Song, and Qi Tian. Spatial coding for large scale partial-duplicate web image search. In: *Proceedings of the 18th ACM international conference on Multimedia.* 2010, pp. 511–520.

[Zho+20] Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 3319–3326.

[Zho+22] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is Geometry Enough for Matching in Visual Localization? In: *European Conference on Computer Vision*. Springer. 2022, pp. 407–425.

[ZK06] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In: *Third international symposium on 3D data processing, visualization, and transmission (3DPVT'06)*. IEEE. 2006, pp. 33–40.

[ZK15] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4353–4361.

[ZR18] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6325–6333.

[ZS10] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In: *European Conference on Computer Vision*. Springer. 2010, pp. 255–268.

[ZSL21] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 4669–4678.

[ZSP15] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2704–2712.

[ZWT99] Zhiqiang Zheng, Han Wang, and Eam Khwang Teoh. Analysis of gray level corner detection. In: *Pattern Recognition Letters* 20.2 (1999), pp. 149–162.