Diss. ETH No. 30319

On Learning and Geometry for Visual Localization and Mapping

A thesis submitted to attain the degree of

Doctor of Sciences

(Dr. sc. ETH Zurich)

presented by

Paul-Edouard Alexandre Sarlin

MSc in Robotics, Systems and Control
ETH Zurich
born on 11.12.1996

accepted on the recommendation of

Prof. Dr. Marc Pollefeys
Prof. Dr. Daniel Cremers
Prof. Dr. Noah Snavely
Dr. Tomasz Malisiewicz

2024

Abstract

Visual localization and mapping are important problems in Computer Vision with widespread use in many applications like Augmented Reality (AR) and Robotics. This problem has been extensively studied in the past decades, resulting in mature solutions based on correspondences across images, well-understood projective geometry, and 3D maps as sparse point clouds. Despite their complexity, such systems struggle with challenges that arise from real-world data. Deep learning offers a promising avenue to address these limitations and reach higher accuracy and robustness.

One strain of research involves replacing specific components of the existing algorithms with Deep Neural Networks (DNNs). While this has led to notable performance improvements, it has also increased system complexity. Additionally, these gains are often constrained because the components are trained with proxy objectives that do not fully capture the ultimate goal of localization. Alternatively, some research has focused on developing simpler black-box DNNs trained end-to-end to replace these complex systems. They have the potential to learn stronger priors but have so far demonstrated limited generalization and interpretability. The balance between generalization and end-to-end training necessitates hybrid algorithms that effectively combine learning capacity with our existing knowledge of 3D geometry.

In the first part of this thesis, we apply this hybrid design philosophy to the prevalent paradigm that is based on 3D maps. We introduce two new algorithms for mapping and localization, both based on the alignment of learned features across different views. To facilitate progress in this research area, we also introduce a new benchmark tailored for AR applications. In the second part, we explore the use of more compact and interpretable 2D maps also used by humans. We demonstrate that end-to-end training enables effectively learning to associate such maps with visual observations. We first develop a new algorithm for localizing images within a 2D semantic map. We then extend our approach to learn a new map representation optimized for visual localization. We introduce an algorithm to construct these 2D maps from visual inputs. Overall, this thesis makes a significant step towards localization and mapping algorithms that integrate robust data-driven priors about the real world.

Résumé

La localisation et la cartograhie à partir d'images sont des problèmes importants dans le domaine de la vision par ordinateur, avec de nombreuses applications telles que la Réalité Augmentée et la robotique. Ce problème a été considérablement étudié au cours des dernières décennies. Cela a abouti à des solutions matures basées sur des correspondences entre images, la géométrie projective, et des carte 3D composées de nuages de points épars. Malgré leur complexité, ces systèmes rencontrent des difficultés face à certaines conditions défavorables que l'on trouve dans le monde réel. L'apprentissage profond offre une voie prometteuse pour surmonter ces limitations et atteindre une plus grande précision et robustesse.

Une approche dans ce domaine de recherche consiste à remplacer certains composants spécifiques des algorithmes existants par des réseaux neuronaux artificiels profonds. Bien que cela ait conduit à des améliorations notables des performances, cela a également augmenté la complexité des systèmes. De plus, ces gains sont souvent limités car les composants sont entraînés avec des objectifs de substitution qui ne capturent pas pleinement l'objectif ultime de la localisation. Alternativement, certaines recherches se sont concentrées sur le développement de réseaux neuronaux plus simples et entraînés de bout en bout pour remplacer ces systèmes complexes. Cette approche peut potentiellement apprendre une modélisation a priori du monde plus expressive. Son interprétabilité et sa capacité de généralisation en dehors de la distribution d'entraînement sont cependant limités. Ce compromis entre généralisation et entraînement de bout en bout requiert des algorithmes hybrides combinent d'une meilleur manière la capacité d'apprentissage avec nos théories existante de géométrie 3D.

Dans la première partie de cette thèse, nous appliquons cette philosophie au paradigme dominant basé sur des cartes 3D. Nous proposons deux nouveaux algorithmes pour la localisation and la cartographie visuelles, tous deux basés sur un alignement des caractéristiques locales au sein de plusieurs images. Pour favoriser les progrès dans ce domaine de recherche, nous proposons un nouveau jeu de données et test de performance spécifiquement conçu pour les applications de Réalité Augmentée. Dans la deuxième partie, nous explorons l'utilisation de cartes 2D simi-

laires à celles utilisées par l'Homme et plus compactes et interprétable que les nuages de points 3D. Nous démontrons que l'entraînement de bout en bout permet d'apprendre efficacement à associer ces cartes avec des observations visuelles. Nous développons d'abord un nouvel algorithme pour localiser des images au sein d'une carte sémantique 2D. Sur cette base, nous étendons notre approche pour apprendre une nouvelle représentation de carte optimisée pour la localisation visuelle. Nous proposons un nouvel algorithme pour construire ces cartes 2D à partir d'observations visuelles. Cumulativement, cette thèse marque une avancée significative vers des algorithmes de localisation et de cartographie basés sur de puissants modèles du monde a priori dérivés des données réelles.

Acknowledgments

My doctoral journey has been a profoundly transformative chapter in my life, made exceptional by the presence of numerous remarkable individuals. This thesis would not have been possible without them.

First and foremost, I am deeply indebted to Marc, my thesis supervisor, for giving me the opportunity to embark on this journey with the Computer Vision and Geometry (CVG) group. The ideal research environment that he fostered, coupled with his encouragement to take risks and explore new avenues, has been invaluable. Marc provided me with the autonomy and freedom I needed to thrive, always believing in my potential and supporting my desire to leave for extended internships. I would also like to thank the external members of my defense committee: Prof. Dr. Daniel Cremers, Prof. Dr. Noah Snavely, and Dr. Tomasz Malisiewicz.

I am fortunate to have collaborated with many brilliant researchers over the years – I am profoundly grateful to all of them. My research journey began at the Autonomous Systems Lab, long before my time at CVG. I am extremely grateful to Marcin, Cesar, and Juan for introducing me to the captivating world of 3D Computer Vision, for sparking my passion for research, and for patiently supporting me when I was a clueless and naive master's student. My time at Magic Leap was also instrumental in the completion of this thesis. My deepest gratitude goes to Tomasz, who significantly shaped the researcher that I am today. Our endless daily discussions taught me invaluable lessons about research and science communication, from selecting promising research topics to delivering memorable presentations and writing papers to be proud of. I am immensely thankful to Daniel, Zak, Vijay, Sri, and the broader team for being highly inspiring and supportive colleagues from whom I learned a great deal.

During my PhD, I had the privilege of interning with several other stellar teams. My sincere thanks to Ondrej, Johannes, and Pablo for warmly welcoming me at Microsoft. Their clarity and advice were crucial when the required efforts felt truly intimidating. Later, I had the chance to visit Meta Reality Labs. Special thanks to Vasileios for giving me the freedom to choose my project, for encouraging me

to take risks, and for putting so much energy into ensuring that the project would continue until completion and publication. I had the privilege of working alongside extraordinary individuals like Peter, Daniel, Tsun-Yi, Armen, Julian, Samuel, and others, whose support and inspiring dedication made my stay unforgettable despite the challenges of remote work. Finally, I had a wonderful time at Google. I am extremely grateful to Simon, Eduard, and Jan for making this opportunity possible and for their continued support. They taught me much and truly inspired me with their dedication, technical expertise, and kindness.

At CVG, I am grateful to have worked with Torsten and Viktor. Their clarity, confidence, and guidance were critical during challenging deadlines. I extend my gratitude to Mihai for our thrilling collaboration on LaMAR and for engaging discussions throughout my time at ETH, covering both research and broader topics. Many thanks to Philipp for our long-lasting exhilarating collaboration and for patiently teaching me so much. His outstanding dedication and deep technical expertise were profoundly inspiring from the beginning. Thanks to Ajay for a fun collaboration and for unwaveringly supporting my odd ideas. Thanks to Zador for our continued collaboration, our captivating discussions, and for reminding me that work should not always be so serious. Finally, I am immensely thankful to Rémi for being an amazing colleague and friend. From implementing SuperPoint to supervising many exciting student projects and embarking on thrilling adventures in the wild, his expertise, kindness, and exemplary work-life balance were true inspirations from which I learned a lot.

Beyond my own research, many exceptional individuals made CVG a very special place. Spending long and late hours at the office never felt like work. My heartfelt gratitude goes to Songyou, Silvan, Mihai, Philipp, Rémi, and Iago. In them, I found long-lasting friends with whom I share many passions in life beyond research. Many thanks to Shaohui, Zador, Viktor, Martin, Zuria, Jonas, and many others for the great time spent together. Thanks also to Ayse for running the lab with such exemplary diligence and relieving us from daunting administrative processes, allowing us to focus on research. I should also thank the fantastic people I met at conferences and summer schools, who have illuminated this journey and made me proud to be part of the Computer Vision community.

To withstand the challenges of research, its highs and lows, many friends have supported me along the way. Foremost, special thanks to the whole 7Up crew, who

were my second family, brightening every day throughout the isolation of the COVID pandemic. They put up with my emotional rollercoasters and showed me a different side of life. I also owe much to Les Gros Chats and to countless others in Zurich, France, and elsewhere, who will recognize themselves here. I am also extremely grateful to Alexandra, my dear partner. In challenging times, she has been incredibly supportive, bringing much-needed stability to my life. She has been exceptionally understanding in the face of the recurring deadlines and never-ending duties of academia, even when they conflicted with much-awaited holidays. Ultimately, she is the one who has rightfully nudged me to maintain a healthier balance. Finally, I am deeply thankful to my beloved parents, Anne and Jean-Jacques, for shaping me into the person I am today. They fed my curiosity about the world from an early age, taught me the values of hard work and perseverance, supported my ventures abroad, but also showed me how to enjoy life.

Paul-Edouard Sarlin
Zurich, September 2024

Contents

CHAPTER 1

Introduction

1.1. Localization and mapping

Image-based localization and mapping are two related problems in photogrammetry and computer vision with widespread use in many applications. Localization, or positioning, is concerned with estimating the location of a device in space given measurements by on-device sensors. The location, or *pose*, is expressed with respect to a given coordinate frame and can be represented by a 2-dimensional (2D) or 3-dimensional (3D) position and possibly with an associated orientation. Localization can be based on diverse technologies and sensors, including radio receivers, inertial measurement units, laser scanners, or cameras. Relying on cameras offers several benefits, as they are inexpensive, can now be found in most consumer devices, and capture rich information about the environment. How to best make use of such information brings new challenges, making image-based localization the focus of this thesis.

Localization often requires a digital map, or representation, of the environment. Humans commonly rely on 2D maps, which represent the Earth's surface, to find their way around. From paper maps to phone applications, such maps are ubiquitous in our daily lives. Examples include planimetric maps, which show the location and spatial arrangement of features, and topographic maps, which show the ground relief and land cover. In the field of computer vision, maps often represent physical properties of the environment in 3D, such as its geometric shape in the form of surfaces. This often relies on computer-friendly data structures like point clouds or meshes. 3D maps often carry additional information, whether it is physical, such as

the appearance of each point in space, in the form of colors, or the material that it is made of, or human-defined, like the semantic classification of each object.

Mapping is the process of estimating a map from raw sensor measurements. In the case of visual mapping, measurements are images taken from different viewpoints and at different points in time. Visual mapping often entails estimating the calibration of the camera, which includes the pose of each image, as position and orientation, and its intrinsic parameters, as zoom level, lens distortion, or color profile. Visual mapping thus often involves the iterative or joint localization of the images.

In the field of computer vision, mapping and localization are hence highly interdependent. In this thesis, we focus on tackling these problem using images only. We first calibrate the *mapping* images and estimate a map from them, which we can later use to localize a new *query* image. Such process can be applied to environments of various scales, such as a single room, a building, a city, or the entire world, with various levels of spatial accuracy in map and camera poses, from millimeter to meter.

1.2. Applications and challenges

Visual localization is useful in applications that require contextual information related to a given location in the environment. In Augmented Reality (AR), the pose of the device should be consistent both across time, to keep virtual content static, and across devices, to share content between users. Similarly, in order to navigate to a given goal, humans and autonomous robots need to know both their starting location and a map of their environment. More generally, the camera calibration estimated by the mapping process are required by multiple downstream computer visions tasks like 3D reconstruction, 3D scene understanding, and novel view synthesis.

These applications bring diverse challenges. Mapping and localization algorithms should be robust to changes that occur in the environment between times at which mapping and query images are taken. This includes appearance changes due to varying illumination conditions at different times of the day, times of the year, or weather. Appearance changes also stem from motion blur due to fast motion and occur when images are taken by camera with different imaging properties.

Differently, structural changes in the 3D shape of the environment stem from moving entities (living beings or motorized objects), cyclic natural phenomena (leaves fall, snow melts), or irreversible changes (buildings are torn down or built).

Some environments are particularly challenging, such as those that exhibit no distinctive texture or objects — white rooms and deserts. The way images are taken also brings challenges, such as large viewpoint differences between the images, in terms of distance to the scene, rotation, or perspective change. Establishing a map from a small number of images with extreme viewpoints is thus significantly more difficult than when a large number of well-distributed images is available. Finally, the scale at which such algorithms should operate can impose tight constraints on their efficiency, in terms of both information storage and computational requirements.

1.3. Background and approaches

Various approaches to visual localization have been studied over the years. One generally distinguishes the problems of place recognition and pose estimation. The former is concerned with estimating a coarse location in a scalable manner. One solution to this is image retrieval, which finds the mapping image that is most similar to the query image [10, 74, 101, 139, 230, 238, 322]. The location accuracy depends on the density of mapping images but these only need to be coarsely located, for example using radio receivers like Global Navigation Satellite System (GNSS). Differently, pose estimation is concerned with extrapolating beyond the set of mapping viewpoints to accurately localize any image. This problem is much more challenging and is the main focus of this thesis. Nonetheless, both problems are often solved together as place recognition can restrict the search space of pose estimation, which is especially beneficial in large-scale environments [138, 198, 233, 261, 262, 308, 323]. This is often called *hierarchical localization*.

Classical 3D geometry: The estimation of the 6-DoF (Degree of Freedom) pose of an image is a long-standing problem in computer vision. It is nowadays tackled by well-established algorithms based on 3D geometry and sparse 3D point clouds. The core building block of these algorithms are *image correspondences*. These are sets of pixels coordinates that correspond to the same 3D point observed in different

images. During the mapping stage, given correspondences between mapping images that observe the same part of the environment, specialized algorithms recover the camera calibration and the 3D coordinates, generally as a sparse 3D point cloud. These algorithms are often based on model fitting [68, 69, 95, 114, 212] and robust least-squares optimization [117, 192, 324]. This process is known as Simultaneous Localization and Mapping (SLAM) or Structure-from-Motion (SfM) depending on the nature of the inputs — SLAM operates on video sequences [78, 152, 206, 213] while SfM requires only unordered images [2, 108, 109, 236, 278, 294, 353]. To later localize a query image, one establishes correspondences between query pixels and 3D points, from which a camera pose is fitted.

Since the camera calibration is a low-dimensional geometric model, one needs only few correspondences to estimate it — 3 correspondences for an absolute pose, 5 correspondences for a relative pose. As such, we generally match only a subset of pixels instead of all of them for computational efficiency. In the most common formulation, correspondences are established between a set of salient points, also called *keypoints*. These are easy to find in different images and well-distributed across each image, and thus often correspond to corners or blobs [116, 175, 183, 220, 251, 253]. Corresponding points are associated using *visual descriptors*. This fingerprint is a high-dimensional vector that encodes the underlying image content in the vicinity of a given point, often by aggregating statistics of the image intensity [11, 29, 43, 183, 253]. Keypoints and descriptors are collectively called local features. Correspondences can also be established implicitly by fitting the geometric model to directly align the image intensity by minimizing a photometric cost [18, 90, 91, 184]. This generally requires a good initial guess, as in SLAM, and is thus less applicable to localization and SfM.

Alternative map representations: Humans often rely on 2D maps to interact with their environments. Examples include planimetric maps, like floorplans for indoors or OpenStreetMap [218] for outdoors, and topographic maps for natural environments. 2D maps can only provide 2D positioning, which is sufficient for many applications since the motion of beings and objects is often constrained to the 2D surface of the Earth because of gravity. However, 2D maps have seen little adoption for image-based machine applications, mostly because it is unclear how to associate their features, which are geometric or semantic, with visual observations. This is unlike LiDAR measurements that are by nature geometric.

Feature-based localization and mapping with 3D sparse point clouds has been extensively studied and developed for decades. It has reached a level of maturity that makes it widely used in various industrial applications. While this problem can be considered solved in the nominal scenarios, existing approaches struggle in the most challenging scenarios described above. The emergence of machine learning for computer vision offers opportunities to tackle them.

Deep learning: A machine learning algorithm can be trained to estimate an output for a new input given a dataset of, in its supervised variant, input-output pairs. Deep Neural Networks (DNNs) are a type of such algorithms that have recently seen the fastest progress and adoption thanks to their impressive capability to learn complex patterns from very large datasets. In the field of computer vision, deep learning was first successfully applied to tasks that involve 2D images. Such tasks include image classification [80, 156, 165], which classifies a foreground object into a set of pre-defined classes, object detection [107, 242], which draws a bounding box around each known object and classifies it, or semantic segmentation, which assigns a semantic class to each pixel in the image [16, 181, 221].

Deep learning has been only later applied to problems involving 3D geometry, like localization and mapping. One strain of research has studied how to improve specific components of the classical pipeline using DNNs. Such components include keypoint detection [27, 81, 85, 245, 330], description [21, 203, 317, 364], matching [177, 263, 365, 371], model fitting [22, 32, 35, 38, 59, 240], and even least-squares optimization [71, 186, 311]. The resulting algorithms often exhibit a higher robustness to challenging viewing conditions by replacing brittle heuristics with more reliable priors learned from the training data. Such priors include, but are not limited to, how appearance changes over time and viewpoints, what keypoints are the most stable and reliable, what are likely motions between two images, or what shapes indoor and outdoor environments often have.

Learned algorithms however bring additional challenges. Since their design and training requires careful implementation and additional expertise, they undoubtedly complexify the localization and mapping pipelines. The lack of interpretability of DNNs also makes it harder to tune the different components such that they fit together, often requiring the re-training of the subsequent ones. One would naturally want to train all components jointly in an end-to-end manner. This is difficult due to the complexity of the entire pipeline and to the fact that some steps, like keypoint

and correspondence selection, are not differentiable or have derivatives that exhibit high variance.

End-to-end learning: A second research strain has studied how to replace this complex pipeline with a simpler DNN trained end-to-end. Mapping algorithms of this kind include DeMon [333], which, given two images, regresses their relative pose and the 3D structure that they observe. Localization algorithms of this kind include DSAC [35], which learns to regress the 3D coordinate associated with each pixel of a given query image. Such approaches have clear benefits: they are simpler algorithms and can learn stronger priors from higher-level supervision. This makes them more robust to challenging viewing conditions and less prone to catastrophic failures.

Black-box neural networks however require large labeled training datasets which are costly to acquire. They exhibit poor generalization outside the training distributions, for example in terms of camera models, viewpoints, types of environments, or visual changes [275]. They are generally less accurate than the classical algorithms and lack flexibility and interpretability — they often cannot leverage partial prior information about camera calibration, poses, or 3D structure. Localization algorithms often even need to implicitly encode the map into their learnable parameters and thus cannot generalize to new scenes [35, 149].

We hypothesize that these drawbacks often stem from a key issue: such approaches generally do not leverage the extensive knowledge of 3D multi-view geometry that we have collectively accumulated through decades of research. This includes the process of image formation, the modeling of lens distortion, or how visual observation relate to quantities like relative poses through epipolar geometry. Neural networks thus need to re-discover this knowledge from scratch each time they are trained. Because these learning algorithms are only statistical, rather than symbolic, they can only approximate such knowledge within the training distribution with no guarantees to generalize well outside of it. How to best leverage 3D geometry within DNNs remains an open problem.

1.4. Contributions

In this thesis, we explore the combination of deep learning and 3D geometry for problems related to visual localization and mapping. We argue that processes for which we have reliable and universal mathematical models, like the formation of an image, should rely on hard geometric rules and not be re-discovered by the learning algorithm. Differently, phenomena that cannot be easily modeled mathematically, like the representation and comparison of appearance or semantics information, should be learned by the algorithms. This requires hybrid DNN architectures that combine geometric and learning components. Optimal data-driven priors should be learned to optimize the end task of localization or mapping. This requires DNNs that can be trained with only high-level supervision by end-to-end backpropagation. As we will see, one way to leverage geometry is as part of an optimization process that is driven by the learned components.

In Part I, we apply this design strategy to the common paradigm that is based on sparse correspondences and 3D point clouds. We derive two new algorithms for mapping and localization that are both based on the alignment of learned features across views. To track the progress in this area of research, we introduce a new benchmark tailored at AR applications.

In Part II, we later note that end-to-end training enables the use of 2D semantic or geometric maps by learning effective representations to associate visual observations — task for which it was previously difficult to design effective heuristics. We first derive a new algorithm to localize an image in 2D semantic maps. We then extend it to learn new map representations that are tailored for visual localization. This requires new algorithms to perform both mapping and localization with such maps.

We summarize the main contributions of this thesis:

Chapter 2: We augment SfM with a refinement process that optimizes 2D keypoints or camera poses and 3D points to align deep features across multiple views. Such features are sufficiently discriminative for a fine-grained alignment but also robust to appearance and viewpoint changes. This refinement process improves the accuracy and robustness of SfM with a small overhead. This chapter is based on work published at ICCV 2021 [176] and TPAMI 2023 [266].

Chapter 3: We leverage this feature alignment process for camera pose refinement. We design an algorithm that learns strong data priors by end-to-end training from pixels to pose and exhibits exceptional generalization to new scenes by separating model parameters and scene geometry. The resulting algorithm learns features that can align images despite long-term appearances changes and dynamic objects. It is surprisingly robust to poor initialization. This chapter is based on work published at CVPR 2021 [268].

Chapter 4: AR is one of the primary applications of visual localization and mapping but research on these problems is still mostly driven by unrealistic benchmarks not representative of real-world AR scenarios. To bridge this gap, we introduce a new dataset and benchmark based on diverse and large-scale scenes recorded with head-mounted and hand-held AR devices along realistic trajectories. The results offer new insights on current research and reveal promising avenues for future work in the field of multi-sensor localization and mapping for AR. This chapter is based on work published at ECCV 2022 [265].

Chapter 5: We make a first step towards leveraging 2D maps for visual localization. As the gravity direction is often measured by inertial sensors, we reduce this problem to estimating a 3-DoF pose. We then derive a learning algorithm that is supervised only by camera poses and learns to perform semantic matching with a wide range of map elements in an end-to-end manner. The algorithm can leverage globally available maps from OpenStreetMap [218], enabling anyone to localize anywhere such maps are available. This chapter is based on work published at CVPR 2023 [264].

Chapter 6: We introduce a framework to learn abstract 2D maps that are optimal for visual localization. We design an algorithm to build such maps from visual observations by both ground-level and aerial cameras. It is trained to align maps estimated from different inputs, supervised only with camera poses over tens of millions of StreetView images. The resulting algorithm can resolve the location of challenging image queries beyond the reach of traditional methods, outperforming the approaches based on 3D point clouds by a large margin. This chapter is based on work published at NeurIPS 2023 [267].

1.5. Publications

This thesis builds from the following publications:

Pixel-Perfect Structure-from-Motion with Featuremetric Refinement
Philipp Lindenberger*, Paul-Edouard Sarlin*, Viktor Larsson, Marc Pollefeys
in *International Conference on Computer Vision (ICCV) 2021*
Best student paper award

Pixel-Perfect Structure-from-Motion with Featuremetric Refinement
Paul-Edouard Sarlin*, Philipp Lindenberger*, Viktor Larsson, Marc Pollefeys
in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2023*

Back to the Feature:
Learning Robust Camera Localization from Pixels to Pose
Paul-Edouard Sarlin*, Ajaykumar Unagar*, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, Torsten Sattler
in *Computer Vision and Pattern Recognition (CVPR) 2021*

LaMAR: Benchmarking Localization and Mapping for Augmented Reality
Paul-Edouard Sarlin*, Mihai Dusmanu*, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, Marc Pollefeys
in *European Conference on Computer Vision (ECCV) 2022*

OrienterNet: Visual Localization in 2D Public Maps with Neural Matching
Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Bulo, Richard Newcombe, Peter Kontschieder, Vasileios Balntas
in *Computer Vision and Pattern Recognition (CVPR) 2023*

SNAP: Self-Supervised Neural Maps
for Visual Positioning and Semantic Understanding
Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, Simon Lynen
in *Advances in Neural Information Processing Systems (NeurIPS) 2023*

The code associated with all publications is publicly available, ensuring that the re-

sults are reproducible. Authors marked by * contributed equally to the corresponding publication.

Throughout the PhD, I also contributed to 3 publications that are not included in this thesis:

LightGlue: Local Feature Matching at Light Speed
Philipp Lindenberger, Paul-Edouard Sarlin, Marc Pollefeys
in *International Conference on Computer Vision (ICCV) 2023*

GeoCalib: Learning Single-image Calibration with Geometric Optimization
Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, Marc Pollefeys
in *European Conference on Computer Vision (ECCV) 2024*

Structure-from-Motion from Pixel-wise Correspondences
Philipp Lindenberger, Paul-Edouard Sarlin, Marc Pollefeys
under review, 2023

Part I.

Localization and Mapping with 3D Maps

CHAPTER 2

Structure-from-Motion with Featuremetric Refinement

Finding local features that are repeatable across multiple views is a cornerstone of sparse 3D reconstruction. The classical image matching paradigm detects keypoints per-image once and for all, which can yield poorly-localized features and propagate large errors to the final geometry. In this chapter, we refine two key steps of structure-from-motion by a direct alignment of low-level image information from multiple views: we first adjust the initial keypoint locations prior to any geometric estimation, and subsequently refine points and camera poses as a post-processing. This refinement is robust to large detection noise and appearance changes, as it optimizes a *featuremetric* error based on dense features predicted by a neural network. This significantly improves the accuracy of camera poses and scene geometry for a wide range of keypoint detectors, challenging viewing conditions, and off-the-shelf deep features. Our system easily scales to large image collections, enabling pixel-perfect crowd-sourced localization at scale.

2.1. Introduction

Sparse or dense 3D reconstructions of the environment can be built from images using Structure-from-Motion (SfM), which associates observations across views to estimate camera parameters and 3D scene geometry. Sparse reconstruction based on matching local image features [30, 81, 85, 116, 183, 226, 245, 263] is the most common due to its scalability and its robustness to appearance changes introduced

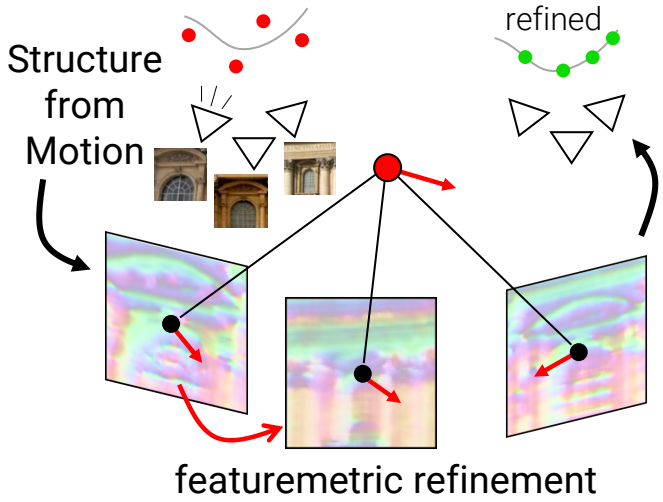


Figure 2.1.: ***From sparse to dense.*** *We improve the accuracy of sparse SfM by refining 2D keypoints, camera poses, and 3D points using the direct alignment of deep features. This featuremetric optimization leverages dense image information but can scale to scenes with thousands of images. Such refinement results in subpixel-accurate reconstructions, even in challenging conditions.*

by varying devices, viewpoints, and temporal conditions found in crowdsourced scenarios [2, 99, 123, 138, 171, 179, 237].

SfM assumes that sparse interest points [30, 81, 85, 116, 183, 245, 252, 330, 364] can be reliably detected across views. It typically selects such points for each image independently and relies on these initial detections for the remainder of the reconstruction process. However, detecting keypoints from a single view is inherently inaccurate due to appearance changes and discrete image sampling [105]. The advent of Convolutional Neural Network (CNN)s for detection has magnified this issue, as they generally do not retain local image information and instead favor global context.

Multi-view geometric optimization with bundle adjustment (BA) [4, 140, 324] is commonly used to refine cameras and points using reprojection errors. Dusmanu et al. [86] proposed to refine keypoint locations prior to SfM via an analogous

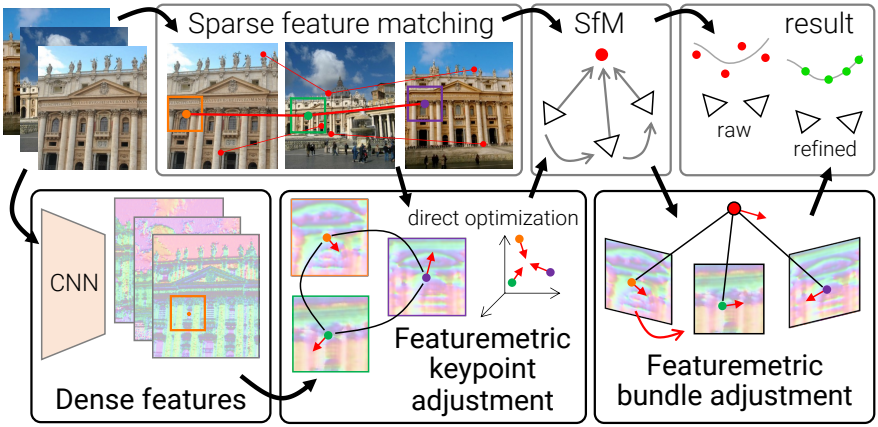


Figure 2.2.: ***Refinement pipeline.*** *Our refinement works on top of any SfM pipeline that is based on local features. We perform a two-stage adjustment of keypoints and bundles. The approach first refines the 2D keypoints only from tentative matches by optimizing a direct cost over dense feature maps. The second stage operates after SfM and refines 3D points and poses with a similar featuremetric cost.*

geometric cost constrained with local optical flow. This can improve SfM, but has limited accuracy and scalability.

In this chapter, we argue that local image information is valuable throughout the SfM process to improve its accuracy. We adjust both keypoints and bundles, before and after reconstruction, by direct image alignment [75, 90, 184] in a learned feature space. Exploiting this locally-dense information is significantly more accurate than geometric optimization, while deep, high-dimensional features extracted by a CNN ensure wider convergence in challenging conditions. This formulation elegantly combines globally-discriminative sparse matching with locally-accurate dense details. It is applicable to both incremental [278, 295] and global [24, 58, 195] SfM irrespective of the types of sparse or dense features.

We validate our approach in experiments evaluating the accuracy of both 3D structure and camera poses in various conditions. We demonstrate drastic improvements for multiple hand-crafted and learned local features using off-the-shelf CNNs. The resulting system produces accurate reconstructions and scales well to large

scenes with thousands of images. In the context of visual localization, it can, in addition to providing a more accurate map, also refine poses of single query images with minimal overhead. We also study in details the sensitivity of the results to various parameters and thus provide insights on how our system can be tuned to fit different use cases and requirements. Finally, we demonstrate that our refinement also improves the accuracy of mapping and localization with (semi-)dense image correspondences and that it scales well to handle denser reconstructions.

For the benefit of the research community, our implementation is freely available as an extension to COLMAP [278] and to the popular localization toolbox hloc [260, 261]. We believe that our featuremetric refinement can significantly improve the accuracy of existing datasets [273] and push the community towards sub-pixel accurate localization at large scale.

2.2. Related work

Image matching is at the core of SfM and visual SLAM, which typically rely on sparse local features for their efficiency and robustness. The process i) detects a small number of interest points, ii) computes their visual descriptors, iii) matches them with a nearest neighbor search, and iv) verifies the matches with two-view epipolar estimation and Random Sample Consensus (RANSAC). The correspondences then serve for relative or absolute pose estimation and 3D triangulation. As keypoints are sparse, small inaccuracies in their locations can result in large errors for the estimated geometric quantities.

Differently, dense matching [64, 178, 247, 285, 303, 320, 327] considers all pixels in each image, resulting in denser and more accurate correspondences. It has been successful for constrained settings like optical flow [137, 302] or stereo depth estimation [362], but is not suitable for large-scale SfM due to its high computational cost due to many redundant correspondences. Several recent works [170, 246, 308, 377] improve the matching efficiency by first matching coarsely and subsequently refining correspondences using a local search. This is however limited to image pairs and thus cannot create point tracks required by SfM. To overcome this limitation, these works group adjacent correspondences using a coarse grid, which impairs the accuracy of SfM.

Our work combines the best of both paradigms by leveraging dense local information to refine sparse observations. It is inherently amenable to SfM as it can optimize all locations over multiple views in a track simultaneously. We show that it can yield highly-accurate reconstructions from even dense but imprecise correspondences.

Subpixel estimation is a well-studied problem in correspondence search. Common approaches either upsample the input images or fit polynomials or Gaussian distributions to local image neighborhoods [98, 125, 134, 183, 277]. With the widespread interest in CNNs for local features, solutions tailored to 2D heatmaps have been recently developed, such as learning fine local sub-heatmaps [132] or estimating subpixel corrections with regression [65, 312] or the soft-argmax [216, 366]. Cleaner heatmaps can also arise from aggregating predictions over multiple virtual views using data augmentation [81].

Detections or local affine frames can be combined across multiple views with known poses in a least-squares geometric optimization [89, 324]. Dusmanu et al. [86] instead refine keypoints solely based on tentative matches, without assuming known geometry. This geometric formulation exhibits remarkable robustness, but is based on a local optical flow whose estimation for each correspondence is expensive and approximate. We unify both keypoint and bundle optimizations into a joint framework that optimizes a featuremetric cost, resulting in more accurate geometries and a more efficient keypoint refinement.

Direct alignment optimizes differences in pixel intensities by implicitly defining correspondences through the motion and geometry. It therefore does not suffer from geometric noise and is naturally subpixel accurate via image interpolation. Direct photometric optimization has been successfully applied to optical flow [18, 184], visual odometry [75, 90, 91, 150], SLAM [6, 282], Multi-View Stereo (MVS) [79, 82, 363], and pose refinement [283]. It generally fails for moderate displacements or appearances changes, and is thus not suitable for large-baseline SfM. One notable work by Woodford & Rosten [354] refines dense SfM+MVS models with a robust image normalization. It focuses on dense mapping with accurate initial poses and moderate appearance changes. Georgel et al. [96] instead estimate more accurate relative poses by elegantly combining photometric and geometric costs. They show that dense information can improve sparse estimation but their approach ignores

appearance changes. Differently, our work improves the entire SfM pipeline starting with tentative matches and addresses larger, challenging changes.

To improve on the weaknesses of photometric optimization, numerous recent works align multi-dimensional image representations. Examples of this *featuremetric* optimization include frame tracking with handcrafted [7, 223] or learned descriptors [71, 186, 339, 340, 357], optical flow [9, 57], MVS [367], and dense SfM in small scenes [311]. Here we extend this paradigm to sparse SfM and propose an efficient algorithm that scales to thousands of images. We show that learning task-specific wide-context features is not necessary and demonstrate highly accurate refinements with off-the-shelf features.

In conclusion, our work is the first to apply robust featuremetric optimization to a large-scale sparse reconstruction problem and show significant benefits for visual localization.

2.3. Background

Given N images $\{\mathbf{I}_i\}$ observing a scene, we are interested in accurately estimating its 3D structure, represented as sparse points $\{\mathbf{P}_j \in \mathbb{R}^3\}$, intrinsic parameters $\{\mathbf{C}_i\}$ of the cameras, and the poses $\{(\mathbf{R}_i, \mathbf{t}_i) \in \mathbf{SE}(3)\}$ of the images, represented as rotation matrices and translation vectors.

A typical SfM pipeline performs geometric estimation from correspondences between sparse 2D keypoints $\{\mathbf{p}_u\}$ observing the same 3D point from different views, collectively called a track. Association between observations is based on matching local image descriptors $\{\mathbf{d}_u \in \mathbb{R}^D\}$, but the estimated geometry relies solely on the location of the keypoints, whose accuracy is thus critical. Keypoints are detected from local image information for each image individually, without considering multiple views simultaneously. Subsequent steps of the pipeline discover additional information about the scene, such as its geometry or its multi-view appearance. Two approaches leverage this information to reduce the detection noise and refine the keypoints.

Global refinement: Bundle adjustment [324] is the gold standard for refining

structure and poses given initial estimates. It minimizes the total geometric error

$$E_{\mathrm{BA}} = \sum_{j} \sum_{(i,u)\in\mathcal{T}(j)} \left\| \Pi\left(\mathbf{R}_i \mathbf{P}_j + \mathbf{t}_i, \mathbf{C}_i\right) - \mathbf{p}_u \right\|_\gamma \quad , \tag{2.1}$$

where $\mathcal{T}(j)$ is the set of images and keypoints in track j, $\Pi(\cdot)$ projects to the image plane, and $\|\cdot\|_\gamma$ is a robust norm [113]. This formulation implicitly refines the keypoints while ensuring their geometric consistency. When available, it can incorporate uncertainties in the location of the initial detections, but often requires many of such observations to reduce the geometric noise. Operating on an existing reconstruction, it cannot recover observations arising from noisy keypoints that are matched correctly but discarded by the geometric verification.

Track refinement: To improve the accuracy of the keypoints prior to any geometric 3D estimation, Dusmanu et al. [86] optimize their locations over tentative tracks formed by raw, unverified matches. They exploit the inherent structure of the matching graph to discard incorrect matches without relying on geometric constraints. Given two-view dense flow fields $\{\mathbf{T}_{v\to u}\}$ between the neighborhoods of matching keypoints u and v, this *keypoint adjustment* optimizes, for each tentative track j, the multi-view cost

$$E^j_{\mathrm{KA}} = \sum_{(u,v)\in\mathcal{M}(j)} \left\| \mathbf{p}_v + \mathbf{T}_{v\to u}[\mathbf{p}_v] - \mathbf{p}_u \right\|_\gamma \quad , \tag{2.2}$$

where $\mathcal{M}(i)$ denotes the set of matches that forms the track and $[\cdot]$ is a lookup with subpixel interpolation. A deep neural network is trained to regress the flow of a single point from two input patches and the flow field is interpolated from a sparse grid. This dramatically improves the keypoint accuracy, but some errors remain as the regression and the interpolation are only approximate.

Both bundle and keypoint adjustments are based on geometric observations, namely keypoint locations and flow, but do not account for their respective uncertainties. They thus require a large number of observations to average out the geometric noise and their accuracy is in practice limited.

2.4. Approach

Summarizing dense image information into sparse points is necessary to perform global data association and optimization at scale. However, refining geometry is an inherently local operation, which, we show, can efficiently benefit from locally-dense pixels. Given constraints provided by coarse but global correspondences or initial 3D geometry, the dense information only needs to be locally accurate and invariant but not globally discriminative. While SfM typically discards image information as early as possible, we instead exploit it in several steps of the process thanks to direct alignment. Leveraging the power of deep features, this translates into featuremetric keypoint and bundle adjustments that elegantly integrate into any SfM pipeline by replacing their geometric counterparts. Figure 2.2 shows an overview.

We first introduce the featuremetric optimization in Sec. 2.4.1. We then describe our formulations of keypoint adjustment (KA), in Sec. 2.4.2, and bundle adjustment, in Sec. 2.4.3, and analyze their efficiency.

2.4.1. Featuremetric optimization

Direct alignment: We consider the error between image intensities at two sparse observations: $\mathbf{r} = \mathbf{I}_i[\mathbf{p}_u] - \mathbf{I}_j[\mathbf{p}_v]$. Local image derivatives implicitly define a flow from one point to the other through a gradient descent update:

$$\mathbf{T}_{v \to u}[\mathbf{p}_v] \propto -\frac{\partial \mathbf{I}_j}{\partial \mathbf{p}}[\mathbf{p}_v]^\top \mathbf{r} \quad . \tag{2.3}$$

This flow can be efficiently computed at any location in a neighborhood around v, without approximate interpolation nor descriptor matching. It naturally emerges from the direct optimization of the photometric error, which can be minimized with second-order methods in the same way as the aforementioned geometric costs. Unlike the flow regressed from a black-box neural network [86], this flow can be made consistent across multiple view by jointly optimizing the cost over all pairs of observations.

Learned representation: SfM can handle image collections with unconstrained viewing conditions exhibiting large changes in terms of illumination, resolution, or camera models. The image representation used should be robust to such changes and ensure an accurate refinement in any condition. We thus turn to features computed by deep CNNs, which can exhibit high invariance by capturing a large context, yet retain fine local details. For each image $\mathbf{I}_i$, we compute a D-dimensional, L2-normalized feature map $\mathbf{F}_i \in \mathbb{R}^{W \times H \times D}$ at identical resolution. We use the same representations for keypoint and bundle adjustments, requiring a single forward pass per image. Our experiments show that multiple off-the-shelf dense local descriptors can result in highly accurate refinements. However, our formulation can also be applied to robust intensity representations, such as the normalized cross-correlation (NCC) over local image patches [354].

2.4.2. Keypoint adjustment

Once local features are detected, described, and matched, we refine the keypoint locations before geometrically verifying the tentative matches.

Track separation: Connected components in the matching graph define tentative tracks – sets of keypoints that are likely to observe the same 3D point, but whose observations have not yet been geometrically verified. Because a 3D point has a single projection on a given image plane, valid tracks cannot contain multiple keypoints detected in the same image. We can leverage this property to efficiently prune out most incorrect matches using the track separation algorithm introduced in [86]. This speeds up the subsequent optimization and reduces the noise in the estimation.

Objective: We then adjust the locations of 2D keypoints belonging to the same track j by optimizing its featuremetric consistency along tentative matches with the cost

$$E^j_{\text{FKA}} = \sum_{(u,v)\in\mathcal{M}(j)} w_{uv} \left\| \mathbf{F}_{I(u)}[\mathbf{p}_u] - \mathbf{F}_{I(v)}[\mathbf{p}_v] \right\|_\gamma \quad , \tag{2.4}$$

where $I(u)$ is the index of the image that contains the keypoint u. w_{uv} is the confidence of the correspondence (u, v), such as the similarity of its local feature descriptors $\mathbf{d}_u^\top \mathbf{d}_v$. This allows the optimization to split tracks connected by weak

correspondences, providing robustness to mismatches. The confidence is not based on the dense features since these are not expected to disambiguate correspondences at the global image level.

Efficiency: This direct formulation simply compares pre-computed features on sparse points and is thus much more scalable than patch flow regression (Eq. (2.2)), which performs a dense local correlation for each correspondence. All tracks are optimized independently, which is very fast in practice despite the sheer number of tentative matches.

Drift: Because of the lack of geometric constraints, the points are free to move anywhere on the underlying 3D surface of the scene. The featuremetric cost biases the updates towards areas with low spatial feature gradients and with better-defined features. This can result in a large drift if not accounted for. Keypoints should however remain repeatable w.r.t. unrefined detections to ensure the matchability of new images, such as for visual localization. It is thus critical to limit the drift, while allowing the refinement of noisier keypoints. For each track, we freeze the location of the keypoint $\bar{u}$ with highest connectivity, as in [86], and constrain the location $\mathbf{p}_u$ of each keypoint w.r.t. to its initial detection $\mathbf{p}_u^0$, such that $\left\|\mathbf{p}_u - \mathbf{p}_u^0\right\| \leq K$.

Once all tracks are refined, the geometric estimation proceeds, typically using two-view epipolar geometric verification followed by incremental or global SfM.

2.4.3. Bundle adjustment

The estimated structure and motion can then be refined with a similar featuremetric cost. Here keypoints are implicitly defined by the projections of the 3D points into the 2D image planes, and only poses and 3D points are optimized.

Objective: We minimize for each track j the error between its observations and a reference appearance $\mathbf{f}^j$:

$$E_{\text{FBA}} = \sum_j \sum_{(i,u)\in\mathcal{T}(j)} \left\|\mathbf{F}_i\left[\Pi\left(\mathbf{R}_i\mathbf{P}_j + \mathbf{t}_i, \mathbf{C}_i\right)\right] - \mathbf{f}^j\right\|_\gamma \quad . \tag{2.5}$$

SfM features ↳ Refinement	indoor						outdoor					
	Accuracy (%)			Completeness (%)			Accuracy (%)			Completeness (%)		
	1cm	2cm	5cm	1cm	2cm	5cm	1cm	2cm	5cm	1cm	2cm	5cm
SIFT [183]	75.62	85.04	92.45	0.21	0.87	3.61	57.64	71.92	85.23	0.06	0.34	2.45
↳ Patch Flow	80.99	89.06	**95.06**	0.24	0.97	**3.88**	64.79	78.90	90.04	**0.08**	0.41	**2.76**
↳ **ours**	**82.82**	**89.77**	94.77	**0.25**	**0.96**	3.75	**68.43**	**80.73**	**91.28**	**0.08**	**0.42**	2.75
SuperPoint [81]	75.76	85.61	93.38	0.59	2.21	8.89	50.45	65.07	80.26	0.10	0.55	3.92
↳ Patch Flow	85.77	91.57	95.85	0.72	2.51	**9.59**	64.94	77.65	88.86	0.15	0.77	4.93
↳ **ours**	**89.33**	**93.58**	**96.58**	**0.74**	**2.53**	9.51	**71.27**	**82.58**	**92.08**	**0.16**	**0.83**	**5.06**
D2-Net [85]	47.18	64.94	83.37	0.47	1.87	7.07	20.87	34.55	56.53	0.03	0.19	1.78
↳ Patch Flow	79.10	86.64	93.26	**1.45**	**4.53**	**12.95**	57.34	70.71	84.12	**0.21**	**1.06**	**6.02**
↳ **ours**	**82.49**	**88.83**	**94.35**	1.36	4.13	11.80	**65.71**	**77.95**	**89.22**	**0.21**	1.01	5.63
R2D2 [245]	66.30	79.21	90.00	0.53	2.06	8.62	49.32	66.10	83.10	0.11	0.55	3.63
↳ Patch Flow	77.94	85.82	92.48	0.66	**2.32**	**9.07**	64.14	78.10	90.18	**0.16**	0.71	**4.09**
↳ **ours**	**80.67**	**87.61**	**93.42**	**0.67**	2.31	8.95	**67.77**	**80.85**	**91.91**	**0.16**	**0.73**	**4.09**

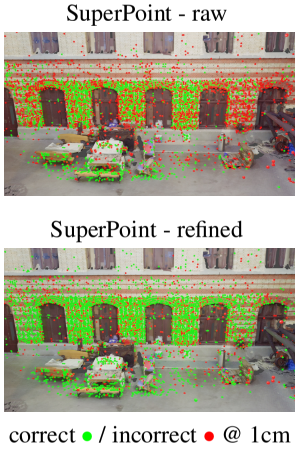


Table 2.1.: ***Evaluation of the 3D sparse triangulation on the ETH3D dataset.*** *Our refinement yields significantly more accurate and complete point clouds than the common geometric SfM pipeline. It is more effective than the existing Patch Flow [86], especially at 1cm or with SIFT. On the left, we show visualizations of the sparse 3D points colored by their accuracy wrt. the ground truth.*

The reference is selected at the beginning of the optimization and kept fixed from then on. This reduces the drift of the points significantly, as also noted in [6], but is more flexible than the common ray-based parametrization [90, 150, 354].

The reference is defined as the observation closest to the robust mean $\boldsymbol{\mu}$ over all initial observations $\mathbf{f}_u^j$ of the track:

$$\mathbf{f}^j = \underset{\mathbf{f} \in \{\mathbf{f}_u^j\}}{\operatorname{argmin}} \left\| \boldsymbol{\mu}^j - \mathbf{f} \right\| \tag{2.6}$$

$$\text{with } \boldsymbol{\mu}^j = \underset{\boldsymbol{\mu} \in \mathbb{R}^D}{\operatorname{argmin}} \sum_{\mathbf{f} \in \{\mathbf{f}_u^j\}} \left\| \mathbf{f} - \boldsymbol{\mu} \right\|_\gamma \quad . \tag{2.7}$$

This ensures robustness to outlier observations and accounts for the unknown topology of the feature space.

Efficiency: Compared to the keypoint adjustment defined in Eq. (2.4), using a reference feature reduces the number of residuals from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. On the other hand, all tracks need to be updated simultaneously because of the interdependency caused by the camera poses. To accelerate the convergence, we form a reduced camera system based on the Schur complement and use embedded point iterations [140]. The refinement generally converges within a few camera updates.

Cost map approximation: Unlike the keypoint adjustment, which can optimize tracks independently, all bundle parameters are updated simultaneously. Given D-dimension features, this involves residuals and Jacobian matrices of dimension D and thus prohibitive memory requirements as often $D{=}128$. Loading from disk all features at each optimization step also incurs large I/O costs. We dramatically increase the efficiency by introducing an approximation based on precomputed distance features.

Given the 2D reprojection $\mathbf{p}_{ij} = \Pi\left(\mathbf{R}_i \mathbf{P}_j + \mathbf{t}_i, \mathbf{C}_i\right)$, the basic formulation loads in memory the dense features $\mathbf{F}_i$, interpolates them at $\mathbf{p}_{ij}$, and compute the residuals $\mathbf{r}_{ij} = \mathbf{F}_i\left[\mathbf{p}_{ij}\right] - \mathbf{f}^j$ for the cost $E_{ij} = \left\|\mathbf{r}_{ij}\right\|_\gamma$. To reduce the memory footprint, we can exhaustively precompute patches of feature distances and treat them as one-dimensional residuals $\bar{\mathbf{r}}_{ij} = \left\|\mathbf{F}_i - \mathbf{f}^j\right\|\left[\mathbf{p}_{ij}\right]$. The cost then becomes $\bar{E}_{ij} = \gamma(\bar{\mathbf{r}}_{ij})$. Such distances only need to be computed once since the reference $\mathbf{f}^j$ is kept fixed throughout the optimization. This precomputed cost reduces the peak memory by a

factor D. It is similar to the Neural Reprojection Error [106] introduced for camera localization.

This approximation displaces the local minimum of the cost by at most 1 pixel but most often by much less. It however degrades the correctness of the approximate Hessian matrix that the Levenberg-Marquardt algorithm [167] relies on for fast convergence. We found that also including the spatial derivatives in the residual significantly improves the convergence. This simply amounts to augmenting the scalar residual map with dense derivative maps:

$$\tilde{\mathbf{r}}_{ij} = \left(\left\| \mathbf{F}_i - \mathbf{f}^j \right\| \quad \frac{\partial \left\| \mathbf{F}_i - \mathbf{f}^j \right\|}{\partial x} \quad \frac{\partial \left\| \mathbf{F}_i - \mathbf{f}^j \right\|}{\partial y} \right)^\top \left[\mathbf{p}_{ij} \right] \quad . \tag{2.8}$$

The approximation thus reduces the residual size from D to 3 with a marginal loss of accuracy.

2.4.4. Implementation

Dense extractor: Our refinement can work with any off-the-shelf CNN that produces feature maps that are locally discriminative. These should be of the same resolution as the input (stride 1) to enable subpixel accuracy. The radius of convergence, or context, of such features depends on the amount of noise in the keypoints. Most detectors like Scale-Invariant Feature Transform (SIFT) [183] have at most a few pixels of error, while others like D2-Net [85] exhibit a much larger detection noise. In our experiments, we use S2DNet [105] for dense feature extraction, as it computes fine features very efficiently in only 4 convolutions, but also produce, if required, deeper features with a larger context. These can then be combined into a multi-level optimization scheme [90, 268, 339] that sequentially refines based on coarse to fine features. The convergence can thus be adjusted depending on the detector and on the image resolution. We show in Sec. 2.5.5 that other dense features work well too.

Optimization: The optimization problems of both keypoint and bundle adjustments are solved with the Levenberg-Marquardt [167] algorithm implemented by the Ceres Solver [3]. Feature maps are stored as collections of 16×16 patches centered around the initial keypoint detections. We thus constrain points to move at most $K = 8$

pixels. The feature lookup is implemented as bicubic interpolation. We use the Cauchy loss γ with a scale of 0.25. The robust mean in Eq. (2.7) is computed with iteratively reweighted least squares [128] initialized with the non-robust mean.

2.5. Experiments

We evaluate our featuremetric refinement on various SfM tasks with several hand-crafted and learned local features and show substantial improvements for all of them. We first evaluate its accuracy on the tasks of triangulation and camera pose estimation in Secs. 2.5.1 and 2.5.2, respectively. We then assess in Sec. 2.5.3 the impact of the refinement on two-view and multi-view pose estimation for end-to-end reconstruction in challenging conditions. We demonstrate in Sec. 2.5.4 that the refinement also scales to and improves dense reconstruction obtained from pixel-wise correspondences. Lastly, Sec. 2.5.5 analyzes the validity and scalability of our design decisions through extensive ablation studies and sensitivity analyses.

2.5.1. 3D triangulation

We first evaluate the accuracy of the refined 3D structure given known camera poses and intrinsics.

Evaluation: We use the ETH3D benchmark [283], which is composed of 13 indoor and outdoor scenes and provides images with millimeter-accurate camera poses and highly-accurate ground truth dense reconstructions obtained with a laser scanner. We follow the protocol introduced in [86], in which a sparse 3D model is triangulated for each scene using COLMAP [278] with fixed camera poses and intrinsics. Following the original benchmark setup, we report the accuracy and completeness of the reconstruction, in %, as the ratio of triangulated and ground-truth dense points that are within a given distance of each other.

Baselines: We evaluate our featuremetric refinement with the hand-crafted local features SIFT [183] and the learned ones SuperPoint [81], D2-Net [85], and R2D2 [245], using the associated publicly available code repositories. We compare our approach to the geometric optimization of [86], referred here as Patch Flow. We

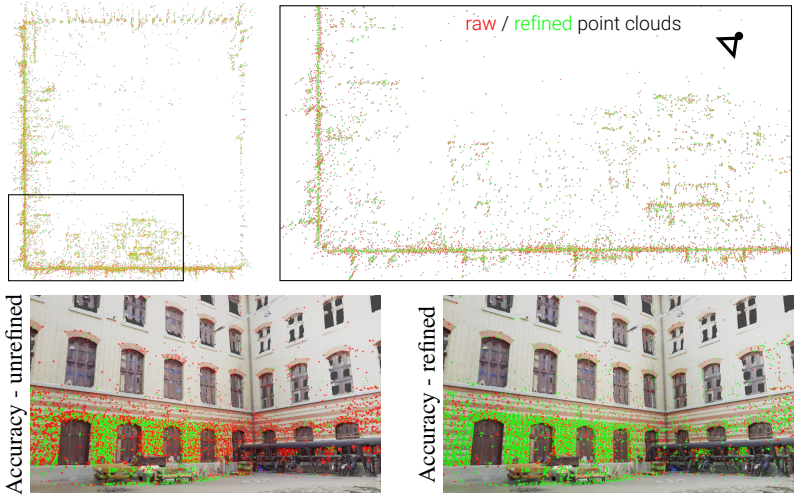


Figure 2.3.: ***Refinement on ETH3D* Courtyard.** *In the top part, we show a top-down view of the sparse point clouds triangulated with raw (in red ●) and refined (in green ●) keypoints. The refined point clouds better fit the geometry of the scene, especially on planar walls. In the lower part, we also show images in which points are colored as accurate (in green ●) or inaccurate (in red ●) at 1cm for raw (left) and refined (right) point clouds.*

re-compute the numbers provided in the original paper using the code provided by the authors.

Results: Table 2.1 shows that our approach results in significantly more accurate and complete 3D reconstructions compared to the traditional geometric SfM. It is more accurate than Patch Flow, especially at the strict threshold of 1cm, and exhibits similar completeness. The improvements are consistent across all local features, both indoors and outdoors. The gap with Patch Flow is especially large for SIFT, which already detects well-localized keypoints. This confirms that our featuremetric optimization better captures low-level image information and yields a finer alignment. Patch Flow is more complete for larger thresholds as it partly solves a different problem by increasing the keypoint repeatability with its large receptive field, while we focus on their localization. We show a qualitative example in Fig. 2.3.

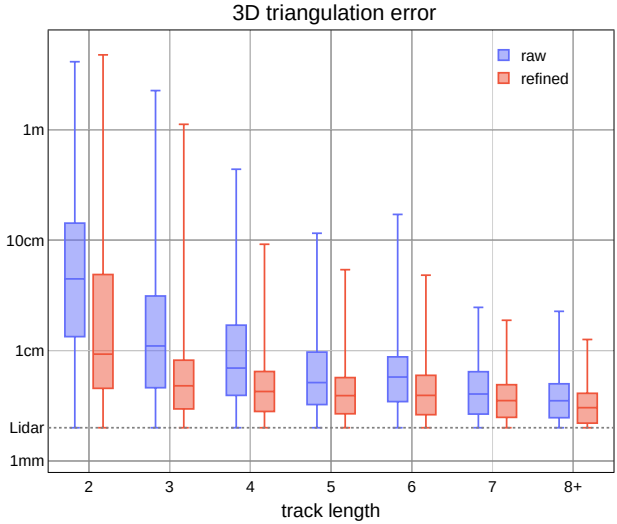


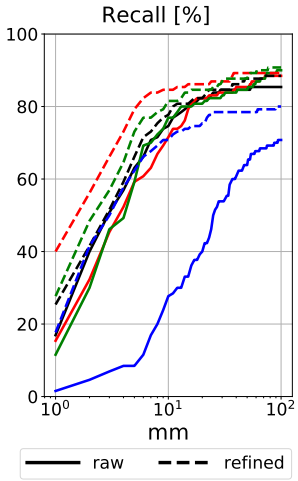
Figure 2.4.: ***Triangulation errors vs. track length.*** *The initial, unrefined output, based on geometric BA, exhibits high errors for 3D points that are observed by few images (low track length). Our refinement significantly reduces these errors and brings the accuracy of the sparse point cloud close to the ground truth acquired by Lidar (2mm accuracy).*

We show in Fig. 2.4 the distribution of triangulation errors for points observed by different numbers of images (track length) for the *Courtyard* scene. Our refinement provides the largest improvement for points with low track length, for which the estimates of the traditional geometric BA are dominated by the noise of the keypoint detection. For larger track lengths, the refined point cloud has an accuracy close to the Faro Focus X 330 laser scanner from which the ground truth is computed.

2.5.2. Camera pose estimation

We now evaluate the impact of our refinement on the task of camera pose estimation from a single image.

Evaluation: We follow the setup of [86] based on the ETH3D benchmark. For each scene, 10 images are randomly selected as queries. For each of them, the



SfM features ↳ Refinement	AUC (%) 1mm	 1cm	 10cm
● SIFT	16.92	56.08	81.65
↳ Patch Flow	14.62	52.69	81.69
↳ **ours**	**25.38**	**60.22**	**84.07**
● SuperPoint	15.38	51.20	82.33
↳ Patch Flow	28.46	63.99	86.79
↳ **ours**	**40.00**	**71.97**	**86.86**
● D2-Net	1.54	12.16	56.10
↳ Patch Flow	16.92	54.70	75.16
↳ **ours**	**17.69**	**55.03**	**76.26**
● R2D2	11.53	52.88	82.69
↳ Patch Flow	25.38	61.42	84.14
↳ **ours**	**27.69**	**63.86**	**86.13**

Table 2.2.: ***Camera pose estimation on the ETH3D dataset.*** *We plot the cumulative translation error (left) and report its AUC (right). Our refinement improves the accuracy of the query camera poses for all local features, even when for SIFT, whose detections are already well-localized. It is generally more accurate than Patch Flow.*

remaining images, excluding the 2 most covisible ones, are used to triangulate a sparse 3D partial model. Each query is then matched against its corresponding partial model and the resulting 2D-3D matches serve to estimate its absolute pose using LO-RANSAC+PnP [68] followed by a non-linear refinement. We compare the 130 estimated query poses to their ground truth and report the area under the curve (AUC) of the cumulative translation error up to 1mm, 1cm, and 10cm.

Refinement: Patch Flow performs multi-view optimization over each partial model independently as well as over the matches between each query and its partial model. Similarly, we first refine each partial model with featuremetric keypoint and bundle adjustments. We then adjust each keypoint in the query image using its tentative 2D-3D correspondences by minimizing the featuremetric error between its observation in the query and the most similar observation of the respective 3D points. Refining the query keypoints before RANSAC increases the number of inlier matches and stabilizes the pose estimation in challenging scenarios where few 3D points are matched.

SuperPoint ↳ Refinement	KA	BA	qKA	qBA	AUC (%)		
					1mm	1cm	10cm
unrefined					15.38	51.20	82.33
↳ refined	✓				16.15	53.34	82.49
↳ refined	✓	✓			16.92	54.71	84.08
↳ refined	✓	✓	✓		38.46	70.44	85.28
↳ **refined (full)**	✓	✓	✓	✓	**40.00**	**71.97**	**86.86**
↳ Patch Flow	✓		✓		28.46	63.04	86.65

Table 2.3.: ***Ablation study for camera pose estimation.*** *The accuracy of the camera pose is improved by refining the map (KA and BA) and by refining the query keypoints before (qKA) and after (qBA) pose estimation. The largest improvement is brought by qKA. It increases the number of inlier matches and the likelihood of finding the correct pose with RANSAC.*

Once an initial pose is estimated with PnP+RANSAC, we refine it via a small featuremetric bundle adjustment over the inlier correspondences. This optimizes each query keypoint against the closest descriptor within the matched track. As opposed to refining each query keypoint against all observations of the track, this has the benefit of scaling linearly in the number of query keypoints and yields a similar accuracy.

Results: The AUC and its cumulative plot are shown in Tab. 2.2. Our refinement substantially improves the localization accuracy for all local features, including SIFT, for which Patch Flow does not show any benefit. At all error thresholds, the featuremetric optimization is consistently more accurate than its geometric counterparts. The accuracy of SuperPoint is raised far higher than other detectors, despite the high sparsity of the 3D models that it produces. This shows how more accurate keypoint detections can result in much more accurate visual localization.

Ablation study: We analyze in Tab. 2.3 how the different kinds of adjustments impact the accuracy of camera localization. All adjustments bring accuracy gains, with the largest ones brought by refining the query keypoints prior to pose estimation.

SfM features (# keypoints) ↳ Refinement	Task 1: Stereo AUC@K°		Task 2: Multiview AUC@5°@N		
	5°	10°	5	10	25
SuperPoint+SuperGlue (2k)	58.78	71.01	63.02	77.36	86.76
↳ **ours**	**65.89**	**76.51**	**68.87**	**82.09**	**89.73**
SIFT (2k)	38.09	48.05	25.12	50.82	77.28
↳ **ours**	**40.59**	**50.87**	**28.01**	**53.59**	**79.49**
D2-Net (4k)	16.83	22.40	16.52	33.07	49.35
↳ **ours**	**25.89**	**33.32**	**21.33**	**40.69**	**57.93**

Table 2.4.: ***End-to-end structure-from-motion on the Phototourism dataset.*** *The refinement improves the accuracy of poses estimated by epipolar geometry (stereo) or a complete SfM pipeline (multiview) with crowd-sourced imagery. Improvements are substantial for both standard (SIFT) and recent (SuperGlue) matching configurations, especially when few images N observe the scene.*

2.5.3. End-to-end Structure-from-Motion

While the previous experiments precisely quantify the accuracy of the refinement, they do not contain any variations of appearance or camera models. We thus turn to crowd-sourced imagery and evaluate the benefits of our featuremetric optimization in an end-to-end reconstruction pipeline.

Evaluation: We use the data, protocol, and code of the 2020 Image Matching Challenge [143, 326]. It is based on large collections of crowd-sourced images depicting popular landmarks around the world. Pseudo ground truth poses are obtained with SfM [278] and used for two tasks. The stereo task evaluates relative poses estimated from image pairs by decomposing their epipolar geometry. This is a critical step of global SfM as it initializes its global optimization. The multiview task runs incremental SfM for small subsets of images, making the SfM problem much harder, and evaluates the final relative poses within each subset. For each task, we report the AUC of the pose error at the threshold of 5°, where the pose error is the maximum of the angular errors in rotation and translation. As the evaluation server accepts at most correspondences, we cannot evaluate our method using the test data. We instead test on a subset of the publicly available validation scenes, and tune the

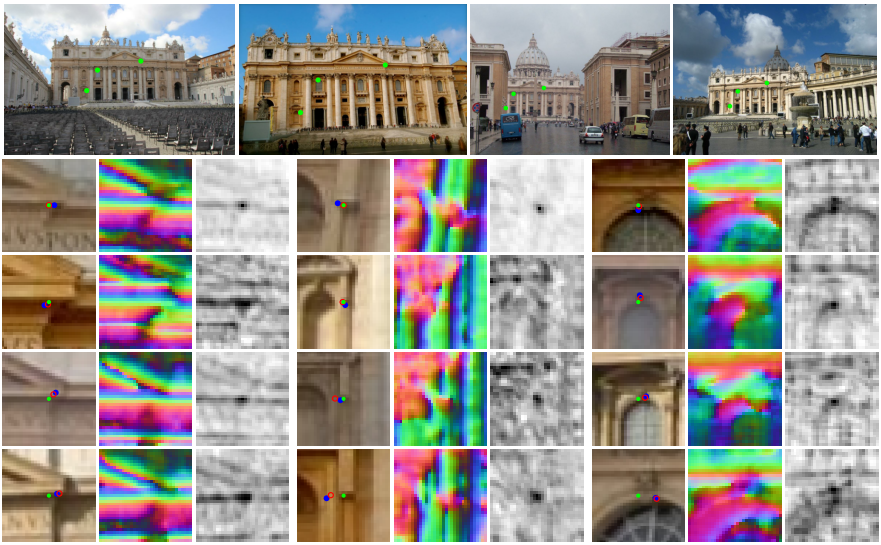


Figure 2.5.: ***Refined SfM tracks.*** *We show patches centered around reprojections of 3x 3D points observed in 4 images of the* St. Peter's Square *scene. Deep features and their correlation maps with a reference are robust to scale or illumination changes, yet preserve local details required for fine alignment. Points refined with our approach (in green ●) are consistent across multiple views while those of a standard SfM pipeline (in red ●) are misaligned because the initial keypoint detections (in blue ●) are noisy.*

RANSAC and matching parameters on the remaining scenes, as recommended by the benchmark.

Baselines: We evaluate our refinement in combination with SIFT [183], D2-Net [85], and SuperPoint+SuperGlue [81, 263]. We limit the number of detected keypoints to 2k for computational reasons, but increase this number to 4k for D2-Net as it otherwise performs poorly. In the stereo task, we adjust the keypoints using the entire exhaustive tentative match graph (4950 pairs per scene). We use LO-DEGENSAC [68, 70] for match verification, the ratio test for SIFT, and the mutual check for SIFT and D2-Net. In the multiview task, we adjust keypoints for each subset independently, considering only the matches between images in the subset, and run our bundle adjustment after SfM.

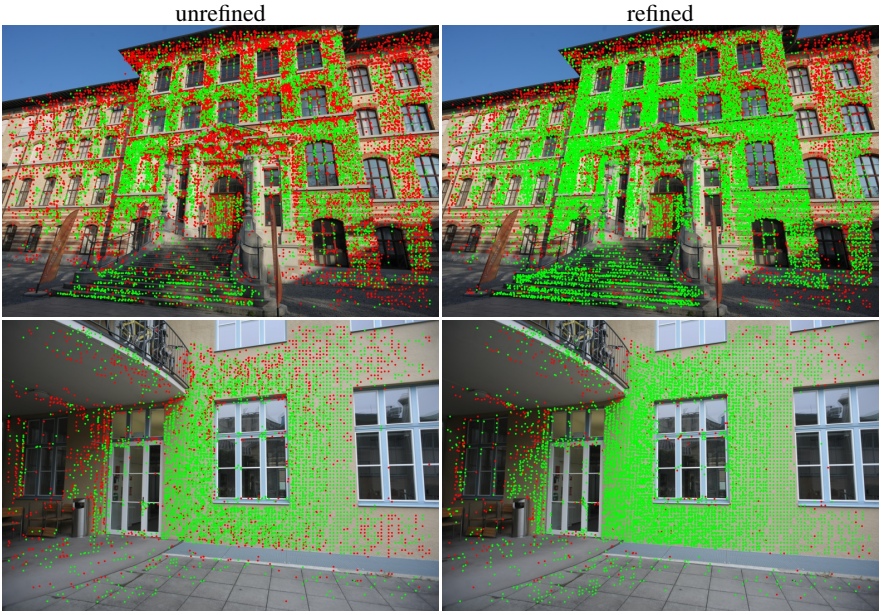


Figure 2.6.: ***Triangulation with dense correspondences.*** *We show reprojections of 3D points triangulated with dense matching for two scenes (top and bottom) reconstructed with (right) and without (left) refinement. The points are colored as accurate (in green ●) or inaccurate (in red ●) wrt. the ground truth for a 1cm error threshold. The refinement yields many more accurate points and can handle extremely dense reconstructions.*

Results: Table 2.4 summarizes the results. For stereo, our featuremetric keypoint adjustment significantly improves the accuracy of the two-view epipolar geometries across all local features and despite the challenging conditions. In multiview setting, it also improves the accuracy of the SfM poses, especially for small sets of images. Featuremetric optimization is particularly effective in this situation, as geometric optimization cannot fully suppress the detection noise due to the small number of observations. We visualize tracks of a 5-image reconstruction in Fig. 2.5 and highlight the accuracy of the refined SfM model.

Discretization ↳ Refinement	triangulation indoor						triangulation outdoor						localization			# observations per image
	Accuracy (%)			Completeness (%)			Accuracy (%)			Completeness (%)			AUC (%)			
	1cm	2cm	5cm	1cm	2cm	5cm	1cm	2cm	5cm	1cm	2cm	5cm	1mm	1cm	10cm	
fine	71.88	83.65	92.60	4.33	12.35	29.95	55.41	71.57	87.49	1.46	6.85	27.26	6.92	47.44	84.81	10844.5
↳ ours	**81.22**	**88.47**	**94.57**	**5.28**	**13.74**	**31.93**	**67.91**	**80.86**	**92.02**	**2.07**	**8.53**	**30.08**	**29.23**	**68.73**	**89.54**	**10968.6**
coarse	53.05	67.45	82.37	0.59	3.07	14.30	33.23	48.11	67.68	0.18	1.13	8.33	2.31	27.65	77.43	960.6
↳ ours	71.01	79.77	88.61	1.44	5.68	20.00	55.16	68.02	81.73	0.44	2.23	12.95	21.54	66.33	89.22	1579.9

Table 2.5.: ***Triangulation and camera pose estimation with dense matching.*** *Our refinement substantially improves the accuracy of triangulation and camera pose estimation when using dense matches discretized with grids of both fine (1px) and coarse (48px) resolutions. The refinement handles well dense reconstructions with a high completeness, which up to 5 times larger than the reconstruction with sparse features reported in Tab. 2.1.*

SuperPoint ↳ Refinement	Acc. (%)		Compl. (%)		AUC
	1cm	2cm	1cm	2cm	1cm
unrefined	18.42	32.23	0.06	0.49	51.20
↳ photometric BA [354]	40.58	60.41	0.17	1.03	68.16
↳ DSIFT [178]	44.67	62.89	0.18	1.06	68.43
↳ VGG16 ImageNet [80, 293]	39.04	57.09	0.16	0.95	68.44
↳ PixLoc [268]	29.49	46.60	0.12	0.74	-
↳ S2DNet [105]	**46.46**	**65.41**	**0.19**	**1.14**	**71.97**

Table 2.6.: ***Triangulation with different dense features.*** *Performing the refinement with different image representations always yields an improvement in triangulation accuracy. S2DNet, our default, works best.*

SuperPoint ↳ Refinement	Acc. (%)		Compl. (%)		Time (s)	Memory (GB)
	1cm	2cm	1cm	2cm		
unrefined	64.27	76.47	0.37	1.44	-	-
↳ ours (exact)	**81.31**	**88.50**	**0.47**	**1.74**	42.22	7.3
↳ ours (cost maps)	80.73	88.48	**0.47**	1.72	**3.33**	0.15
↳ ours (no derivatives)	79.05	87.17	0.46	1.71	16.12	**0.05**

Table 2.7.: ***Triangulation with cost map approximations.*** *Using precomputed cost maps increases the time and memory efficiency of the bundle adjustment with a marginal loss of accuracy and completeness. Only optimizing the cost but not its derivatives saves more memory but yields reconstructions of lower quality with slower convergence.*

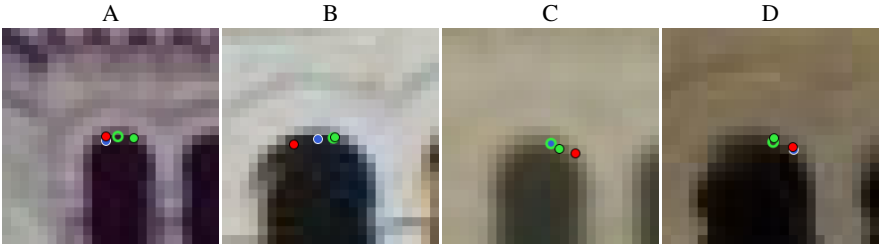


Figure 2.7.: ***The keypoint adjustment increases the track length.*** *We show image patches centered around four observations of the same 3D point. The initial detected keypoints ● are all matched together. Due to the detection noise, the triangulation splits them into two short tracks (A-B and C-D) with high reprojection errors ●. The keypoint adjustment reduces the detection noise ○ and produces a single longer track with a low reprojection error ●.*

2.5.4. Dense image matching

We show that our refinement is also applicable to SfM with dense image matches without any prior keypoint detection. This can improve the robustness of the localization in situations for which detecting repeatable keypoints is challenging, such as low illumination or textureless scenes.

Setup: Several recent works on dense matching [303, 328, 377] demonstrate that they are also amenable to SfM with existing pipelines [278]. These works produce, for each image pair, a set of correspondences between arbitrary subpixel image coordinates, generally arranged as a semi-dense grid. Since SfM requires tracks with a sufficient number of observations, these works partition each image into a grid and assign to the same tracks all pairwise correspondences associated with the same cells. This discretizes arbitrary image coordinates into the center of each cell. A coarser grid reduces the number of tracks and allows the SfM process to scale to large scenes, but the discretization introduces large errors in the estimated geometry. Following the track building, we can run our keypoint and bundle adjustments in the same way as for sparse correspondences. We evaluate triangulation and camera pose estimation as in Secs. 2.5.1 and 2.5.2 by estimating correspondences with LoFTR [303].

	SuperPoint ↳ Refinement	Acc. (%)		Compl. (%)		track length	AUC
		1cm	2cm	1cm	2cm		1cm
KA vs. BA	unrefined	18.42	32.23	0.06	0.49	4.17	51.20
	↳ Patch Flow [86]	37.00	55.18	0.15	0.93	**5.24**	63.53
	↳ F-KA	36.85	54.48	0.15	0.90	5.02	69.84
	↳ F-BA	43.65	62.44	0.18	1.06	4.17	67.61
	↳ **F-KA+BA (full)**	**46.46**	**65.41**	**0.19**	**1.14**	5.02	**71.97**
bonus	w/ F-BA drift	47.93	66.52	0.20	1.17	5.02	64.51
	Patch Flow + F-BA	46.30	65.22	0.19	1.13	5.24	-
	higher resolution	47.67	65.39	0.21	1.21	5.12	73.42

Table 2.8.: ***Ablation study on the ETH3D dataset.*** *i) Featuremetric keypoint and bundle adjustments (KA and BA) both largely improve the triangulation (acc., completeness, track length) and the localization accuracy (AUC). Patch Flow produces a longer track length because of its larger receptive field but is less accurate. ii) Letting the BA drift by updating reference features or increasing the image resolution both improve the triangulation, at the expense of poorer localization and increased run time, respectively.*

Results: We show the results in Tab. 2.5 for two level of discretization, fine and coarse. The refinement improves on all metrics for both triangulation and localization. It makes dense matching competitive with sparse features and even yields more accurate camera poses at 10cm. Given such a large number of observations and correspondences, PatchFlow would require prohibitively high run-times. Our refinement handles well this high density. We show in Fig. 2.6 visualizations of the 3D points triangulated and colored by their accuracy at 1cm. The density of 3D points is significantly higher than regular sparse keypoints.

2.5.5. Additional insights

Ablation study: Table 2.8 shows the performance of variants of our featuremetric optimization on ETH3D in terms of triangulation (scene *Facade* only) and localization (all scenes). Both keypoint and bundle adjustments bring improvements across all metrics. Minor tuning can further improve some dimensions with trade-offs. The keypoint adjustment is particularly effective at increasing the average track length via the number of inlier correspondences. We show an example in Fig. 2.7.

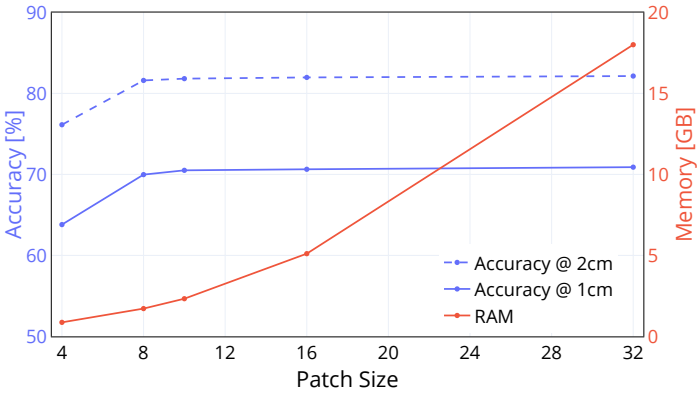


Figure 2.8.: ***Impact of the patch size.*** *Smaller patches for each observation significantly reduce memory requirements but can impair the accuracy of the refinement. Patches of size 10×10 offer a good trade-off with high accuracy and moderate memory consumption.*

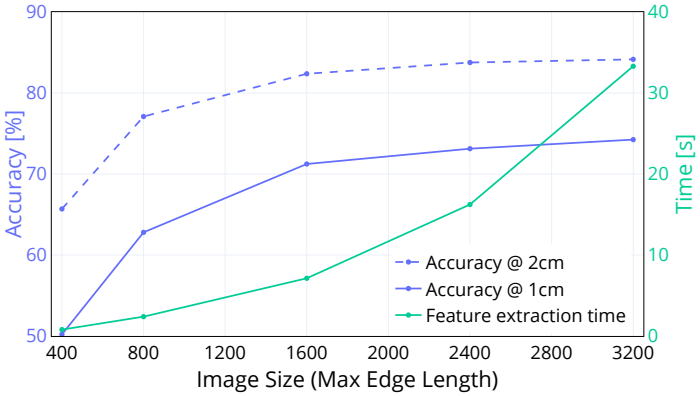


Figure 2.9.: ***Impact of the image resolution.*** *Increasing the image resolution increases the triangulation accuracy, but at the cost of longer feature extraction time and higher VRAM requirements. For all experiments on ETH3D, we used a maximum edge length of 1600px, which is very close to saturating the accuracy while providing low run times.*

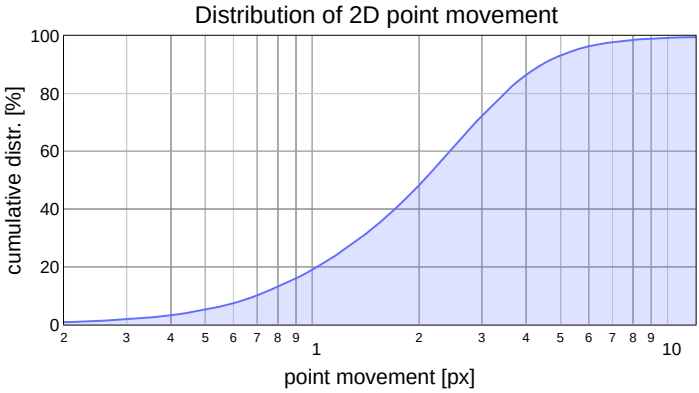


Figure 2.10.: ***Distribution of point movements.*** *We show the cumulative distribution of the distance traveled by the 2D keypoints during the featuremetric refinement of SuperPoint with KA and BA. 60% of the points move by fewer than 2 pixels and 99% remain within 8 pixels of the initial detections.*

Patch size: Figure 2.10 shows how much our refinement displaces the detected keypoints during the triangulation of SuperPoint on *Courtyard* using dense features extracted from 1600×1066-pixel images. When using full feature maps without any constraints in keypoint adjustment, most points are moved by more than 1 pixel, but most often by less than 8 pixels. This confirms that storing the feature maps as 16×16 patches is sufficient and rather conservative. We show in Fig. 2.8 the accuracy of the triangulation for various patch sizes. Smaller 10×10 patches achieve sufficient accuracy and require significantly less memory.

Image resolution The image resolution at which the dense features are extracted has a large impact on the accuracy of the refinement. In Fig. 2.9 we quantify the impact on both triangulation accuracy and run time for the ETH3D *Courtyard* scene (38 images) using an NVIDIA RTX 1080 Ti GPU. The accuracy drops significantly when the resolution is smaller than 1600×1066px, which amounts to 25% of the full image resolution. Doubling the resolution to 3200×2132px yields noticeable improvements, albeit significantly increases the extraction time and the consumption of GPU VRAM. For a resolution of 1600×1066px, as used for all experiments on ETH3D, S2DNet can extract features for 5 images per second. As a reference,

extracting only fine-level S2DNet features (4 convolutions) from 3200×2132px images requires around 10GB of GPU VRAM.

Dense features: We evaluate our refinement with different image representations, including NCC-normalized intensity patches with fronto-parallel warping. We show the results for triangulation in Tab. 2.6. Our final configuration, based on on the dense features of S2DNet [105], performs best across all metrics.

Relying on compact features would easily reduce the memory footprint and the run time of the refinement. To demonstrate these benefits, we show in Fig. 2.11 the relationship between the dimension, the run time of the BA, and the triangulation accuracy when retaining only the first k channels of the S2DNet features. Features with fewer dimensions yield a faster refinement. The accuracy drops moderately but we expect a smaller reduction with features explicitly trained for smaller dimensions.

Cost map approximation: We show experimentally that this approximation often does not, or only minimally, impairs the accuracy of the refinement. Table 2.7 reports the results of the triangulation of SuperPoint features on the ETH3D dataset averaged over indoor and outdoor scenes. The approximation reduces the accuracy by less than 1% and does not alter the completeness. It however significantly reduces the memory consumption of the bundle adjustment. Only optimizing the feature distance yields a drop in accuracy, which is resolved by including the derivatives. Note that the approximation is disabled in all previous experiments as the corresponding scenes are sufficiently small.

Scalability: We run SfM on subsets of images of the Aachen Day-Night dataset [273, 274, 372]. Figure 2.12 shows the run times of the refinement for subsets of 10, 100 and 1000 images and Fig. 2.13 shows the contribution of each step. The featuremetric refinement is one to two orders of magnitude faster than Patch-Flow [86]. Precomputing cost maps reduces the peak memory requirement of the bundle adjustment from 80GB to less than 10GB for 1000 images. As storing feature maps only requires 50GB of disk space, this refinement can easily run on a desktop PC. We thus refined the entire Aachen Day-Night v1.1 model, composed of 7k images, in less than 2 hours. Scene partitioning [278] could further reduce the peak memory.

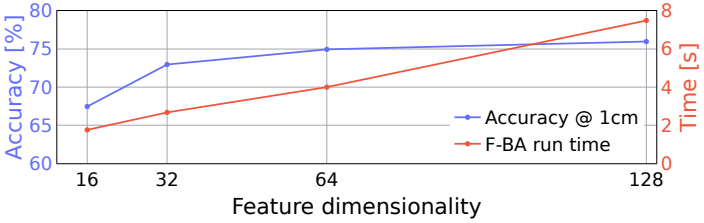


Figure 2.11.: ***Impact of the feature dimensionality.*** *Dense features computed by S2DNet can be naively reduced to accelerate the featuremetric BA by 2 while incurring only a minor drop of triangulation accuracy.*

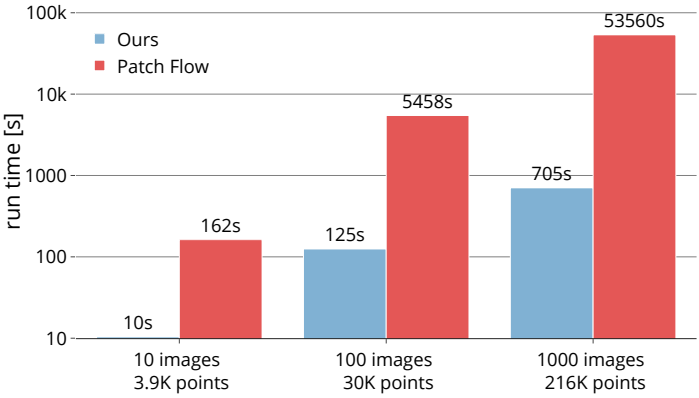


Figure 2.12.: ***Run time depending on the number of images.*** *We show the duration, in logarithmic scale, of the refinement for varying numbers of images. Our refinement is more than ten times faster than Patch Flow [86], whose run-time is dominated by the computation of the pairwise flow, which scales quadratically.*

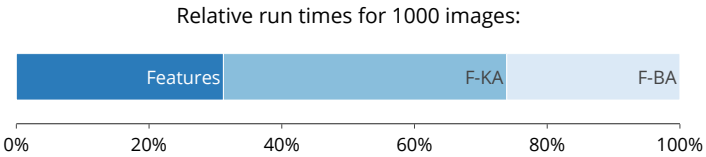


Figure 2.13.: ***Contribution of each step to the run time.*** *Thanks to our precomputed cost patches, the featuremetric BA is fast. The KA amounts for the majority of the refinement time.*

2.6. Summary and outlook

Summary: In this chapter we argue that the recipe for accurate large-scale SfM is to perform an initial coarse estimation using sparse local features, which are by necessity globally-discriminative, followed by a refinement using locally-accurate dense features. Since the dense feature only need to be locally-discriminative, they can afford to capture much lower-level texture, leading to more accurate correspondences. Through extensive experiments we show that this results in more accurate camera poses and structure; in challenging conditions and for different local features.

While we optimize against dense feature maps, we keep the sparse scene representation of SfM. This ensures not only that the approach is scalable but also that the resulting 3D model is compatible with downstream applications, e.g., mapping for visual localization. Since our refinement works well even with few observations, as it does not need to average out the keypoint detection noise, it has the potential to achieve more accurate results using fewer images.

We thus believe that our approach can improve the accuracy of the ground truth poses of standard benchmark datasets, of which many are currently saturated. Since this refinement is less sensitive to under-sampling, it enables benchmarking for crowd-sourced scenarios beyond densely-photographed tourism landmarks.

Impact and limitations: While we have shown that this refinement can scale to thousand of images, its memory footprint, in terms of both disk and RAM, is significantly higher than conventional SfM. This limitations mainly stem from the high dimensionality of the features employed. We believe that training more compact features would make the refinement much more effective and incur only a small drop of accuracy. This has been a clear obstacle to a wider adoption and prohibits its integration in resource-constrained applications. Nevertheless, our work has been found particularly useful by the research community interested in novel view synthesis, which includes approaches like neural radiance fields [200], relies on SfM to obtain camera calibration, and is highly sensitive to calibration errors [329]. Several follow-up works [121, 346, 370] have confirmed that our refinement is highly beneficial in the sparse view setting beyond the datasets considered in our evaluation. We have however found that the proposed refinement yields little to

no improvements when views are densely sampled with high visual overlap, such as for data recorded as videos. This scenario is unfortunately not covered by our evaluation.

Additionally, because the refinement is based on a fairly complex system and requires non-trivial engineering, we have so far not seen any follow-up work build on top of and extend our approach. Lastly, the refinement is only partially robust to appearance changes, thanks to the off-the-shelf features that we employ. Extreme changes severely degrade the accuracy of the refinement. This scenario is unfortunately not covered by our evaluation due to the lack of datasets that include both long-term changes and highly-accurate ground truth. In Chapter 4, we introduce a new dataset and benchmark to bridge this gap. To increase the robustness to appearance changes, we study next in Chapter 3 how such features can be trained end-to-end, at the cost of a lower pixel-level accuracy.

CHAPTER 3

Learning Robust Camera Localization from Pixels to Pose

Camera pose estimation in known scenes is a 3D geometry task recently tackled by multiple learning algorithms. Many regress precise geometric quantities, like poses or 3D points, from an input image. This either fails to generalize to new viewpoints or ties the model parameters to a specific scene. In this chapter, we go Back to the Feature: we argue that deep networks should focus on learning robust and invariant visual features, while the geometric estimation should be left to principled algorithms. We introduce PixLoc, a scene-agnostic neural network that estimates an accurate 6-DoF pose from an image and a 3D model. Inspired by the results presented in Chapter 2, our approach is based on the direct alignment of multiscale deep features, casting camera localization as metric learning. PixLoc learns strong data priors by end-to-end training from pixels to pose and exhibits exceptional generalization to new scenes by separating model parameters and scene geometry. The system can localize in large environments given coarse pose priors but also improve the accuracy of sparse feature matching by jointly refining keypoints and poses with little overhead.

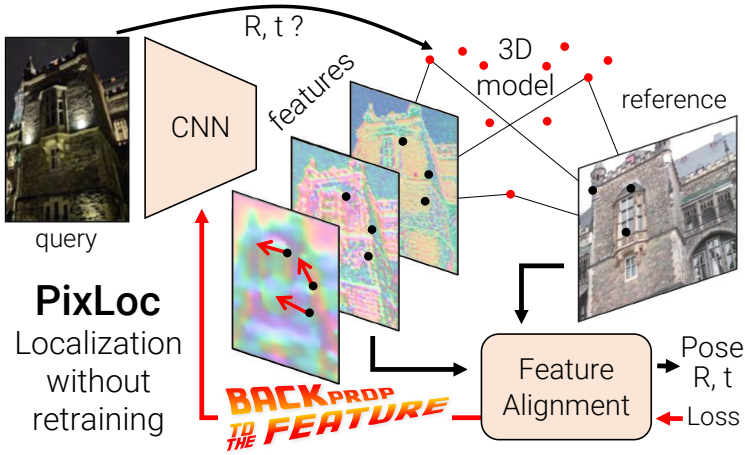


Figure 3.1.: ***Learning scene-agnostic localization.*** *Deep neural networks should not have to rediscover well-understood geometric principles. We only need to learn good features: PixLoc is trained end-to-end to estimate the pose of an image by aligning deep features with a reference 3D model via a differentiable optimization.*

3.1. Introduction

State-of-the-art approaches to visual localization commonly rely on correspondences between 2D pixel positions and 3D points in the scene [39, 52, 105, 187, 261, 263, 272, 306, 308, 318]. Such a formulation estimates the camera pose using a Perspective-n-Point (PnP) solver [5, 40, 114, 154, 157] inside a RANSAC loop [23, 69, 95, 164]. These 2D-3D correspondences are traditionally computed by matching local image features. Recent localization systems can handle large scenes with complex geometry and appearance changes over time. They leverage deep neural networks that learn to extract such features [32, 81, 85, 226, 245, 280, 361], to match them [226, 263], and to filter outlier correspondences [38, 161, 204, 263, 319].

Training a feature matching pipeline in an end-to-end manner is challenging and unstable as its complexity hinders gradients propagation [32]. An alternative is to train a Deep Neural Network (DNN) to regress geometric quantities such as camera poses [20, 83, 147, 149, 163, 342, 378] or the 3D scene coordinate corresponding

to each pixel [35–37, 39, 52, 53, 168, 291, 360]. While these approaches can be trained end-to-end, they come with their own drawbacks. Absolute pose and coordinate regression are scene-specific and require to be trained for or adapted to new scenes [52, 53]. Generalization to new viewing conditions, e.g., localizing night-time images when training only on daytime photos, and handling larger, more complex scenes [280, 308] are open challenges for such approaches. Additionally, absolute or relative pose regression has limited accuracy and often fails to generalize to new viewpoints [275, 378]. While regressing poses relative to a set of reference images [20, 83, 163, 378] is in theory scene-agnostic, generalization to strongly differing scenes without a significant drop in pose accuracy [275, 378] has, to the best of our knowledge, not been shown so far.

What hinders the generalization of existing end-to-end regression methods is that they predict camera poses or 3D geometry solely from image information. In practice, such quantities are often readily available. Pose priors can be obtained via image retrieval or sensors such as GNSS. At the same time, the 3D scene geometry is often provided as a by-product of the 3D reconstruction systems that generate the training poses, e.g., with SfM or SLAM.

Inspired by direct image alignment [75, 90, 91, 223, 339, 340] and learned image representations for outlier rejection [161], we argue that end-to-end visual localization algorithms should focus on representation learning. Rather than devoting model capacity and data to learn basic geometric relations or encode 3D maps, they should rely on well-understood geometric principles and instead learn robustness to appearance and structural changes.

In this chapter, we introduce a trainable algorithm, PixLoc, that localizes an image by aligning it to an explicit 3D model of the scene based on dense features extracted by a Convolutional Neural Network (CNN) (Fig. 3.1). By relying on classical geometric optimization, the network does not need to learn pose regression itself, but only to extract suitable features, making the algorithm accurate and scene-agnostic. We train PixLoc end-to-end, from pixels to pose, by unrolling the direct alignment and supervising only the pose. Given an initial pose obtained by image retrieval, our formulation results in a simple localization pipeline competitive with complex state-of-the-art approaches, even when the latter are trained specifically per scene. PixLoc can also refine poses estimated by any existing approach as a lightweight post-processing step. Through detailed experiments, we show that our

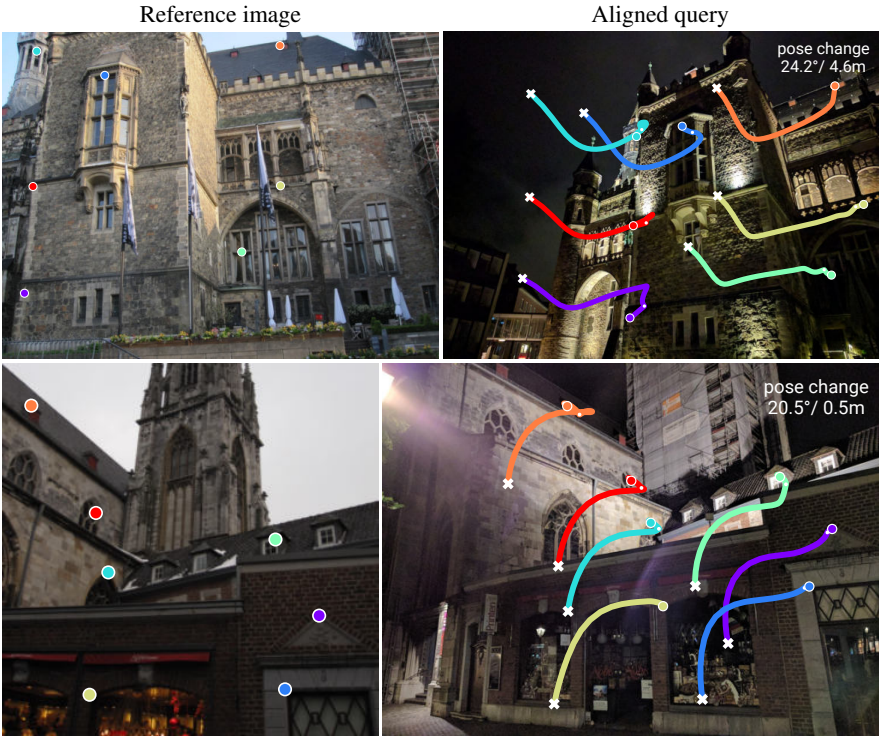


Figure 3.2.: ***Alignment for localization.*** *Although only based on local gradients, direct alignment works well thanks to deep features, despite the coarse initial pose estimate and strong appearance changes. Here points travel from crosses to colored dots.*

method generalizes well to new scenes, e.g., from outdoor to indoor scenes, and challenging viewing conditions. To the best of our knowledge, PixLoc is the first end-to-end visual localization approach to exhibit such exceptional generalization.

3.2. Related work

Accurate visual localization commonly relies on estimating correspondences between 2D pixel positions and 3D scene coordinates. Such approaches detect,

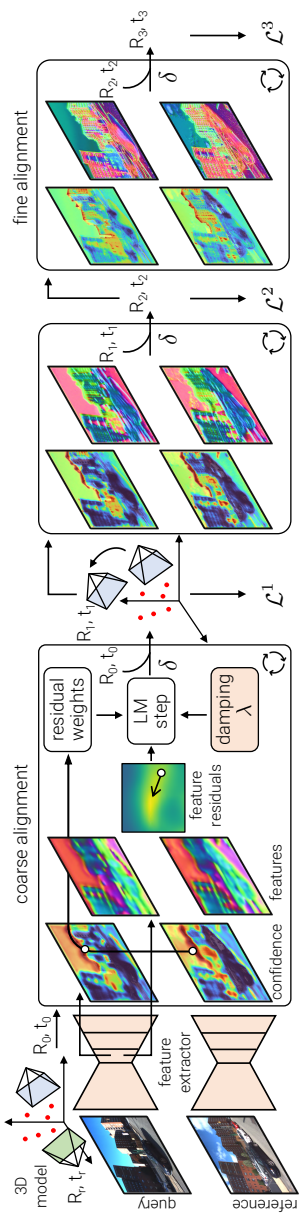


Figure 3.3.: *Pose estimation with PixLoc.* *Given a sparse 3D model and a coarse initial pose $(\mathbf{R}_0, \mathbf{t}_0)$, PixLoc extracts multilevel features with pixelwise confidences for query and reference images. The Levenberg-Marquardt optimization then aligns corresponding features according to the 3D points, guided by the confidence, from the coarse to the fine level. We only supervise the pose predicted at each level.*

describe [30, 183], and match [138, 171, 179, 272, 306, 368] local features, maintain an explicit sparse 3D representation of the environment, and sometimes leverage image retrieval [139, 322] to scale to large scenes [138, 198, 261, 274, 308, 323]. Recently, many of these components have been learned with great success [10, 81, 85, 204, 214, 238, 245, 263, 364], but often independently and not end-to-end due to the complexity of such systems. Here we introduce a simpler alternative to feature matching, finally enabling stable end-to-end training. Our solution can learn more powerful priors than individual blocks, yet remains highly flexible and interpretable.

End-to-end learning for localization has recently received much attention. Common approaches encode the scene into a deep network by regressing from an input image to an absolute pose [147, 149, 209, 239, 342] or 3D scene coordinates [35, 39, 52, 53, 291]. Pose regression lacks geometric constraints and thus does not generalize well to novel viewpoints or appearances [275, 280], while coordinate regression is more robust. Both do not scale well due to the limited network capacity [37, 308] and require for each new scene either costly retraining or adaptation [52, 53]. ESAC [37] improves the scalability by training an ensemble of regressors, each specialized in a scene subset, but is still significantly less accurate than feature-based methods in larger environments.

Differently, some approaches regress a camera pose relative to one or more training images [20, 83, 163, 378], often after an explicit retrieval step. They do not memorize the scene geometry and are thus scene-agnostic, but, similar to absolute regressors, are less accurate than feature-based methods [275, 378]. Closer to ours, SANet [360] takes the scene representation out of the network by regressing 3D coordinates from an input 3D point cloud. Critically, all top-performing learnable approaches are at least trained per-dataset, if not per-scene, and are limited to small environments [149, 291]. In this chapter, we demonstrate the first end-to-end learnable network that generalizes across scenes, including from outdoor to indoor, and that delivers performance competitive with complex pipelines on large real-world datasets, thanks to a differentiable pose solver.

Learning camera pose optimization can be tackled by unrolling the optimizer for a fixed number of steps [71, 186, 190, 311, 344, 357], computing implicit derivatives [39, 44, 59, 145, 256], or crafting losses to mimic optimization steps [339, 340]. Multiple works have proposed to learn components of these optimizers [71, 186, 311],

with added complexity and unclear generalization. Some of these formulations optimize reprojection errors over sparse points, while others use direct objectives for (semi-)dense image alignment. The latter are attractive for their simplicity and accuracy, but usually do not scale well. Like their classical counterparts [90, 150], they also suffer from a small basin of convergence, limiting them to frame tracking or point refinement, as seen in Chapter 2. In contrast, PixLoc is explicitly trained for wide-baseline cross-condition camera pose estimation from sparse measurements (Fig. 3.2). By focusing on learning good features, it shows good generalization yet learns sensible data priors that shape the optimization objective.

3.3. PixLoc: from pixels to pose

Overview: PixLoc localizes by aligning query and reference images according to the known 3D structure of the scene. The alignment consists of a few steps that minimize an error over deep features predicted from the input images by a CNN (Fig. 3.3). The CNN and the optimization parameters are trained end-to-end from ground truth poses.

Motivation: In absolute pose and scene coordinate regression from a single image, a deep neural network learns to i) recognize the approximate location in a scene, ii) recognize robust visual features tailored to this scene, and iii) regress accurate geometric quantities like pose or coordinates. Since CNNs can learn features that generalize well across appearances and geometries, i) and ii) do not need to be tied to a specific scene, and i) is already solved by image retrieval. On the other hand, iii) is tackled by classical geometry using feature matching [68, 69, 95] or image alignment [18, 90, 91, 184] and a 3D representation. We should thus focus on learning robust and generic features, making the pose estimation scene-agnostic and tightly constrained by geometry. The challenge lies in how to define good features to localize. We solve this by making the geometric estimation differentiable and supervise only the final pose estimate. Differently from pose or coordinate regression, we assume that a 3D scene representation is available. This requirement is easily met in practice since the reference poses are usually obtained by sparse or dense 3D reconstruction.

Problem formulation: Our goal is to estimate the 6-DoF pose $(\mathbf{R}, \mathbf{t}) \in \mathbf{SE}(3)$ of a query image $\mathbf{I}_q$, where $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is a translation vector in the camera frame. We are given a 3D representation of the environment, such as a sparse or dense 3D point cloud $\{\mathbf{P}_i\}$ and posed reference images $\{\mathbf{I}_k\}$, collectively called the reference data.

3.3.1. Localization as image alignment

Image Representation: The sparse alignment is performed over learned feature representations of the images. We leverage CNNs and their ability to extract a hierarchy of features at multiple levels. For each query image $\mathbf{I}_q$ and reference image $\mathbf{I}_k$, a CNN extracts a D_l-dimensional feature map $\mathbf{F}^l \in \mathbb{R}^{W_l \times H_l \times D_l}$ at each level $l \in \{L, \dots, 1\}$. Those have decreasing resolution and progressively encode richer semantic information and a larger spatial context of the image. The features are L_2-normalized along the channels to improve their robustness and generalization across datasets.

This learned representation, inspired by past works on handcrafted and learned features for camera tracking [75, 186, 223, 311, 339, 344], is robust to large illumination or viewpoint changes and provides meaningful gradients for successful alignments despite poor initial pose estimates. In contrast, classical direct alignment [18, 90, 91, 184] operates on the original image intensity, which is not robust to long-term changes encountered in common localization scenarios, and resorts to Gaussian image pyramids, which still largely limits the convergence to frame-to-frame tracking.

Direct alignment: The goal of the geometric optimization is to find the pose $(\mathbf{R}, \mathbf{t})$ which minimizes the difference in appearance between the query image and each reference image. For a given feature level l and each 3D point i observed in each reference image k, we define a residual:

$$\mathbf{r}_k^i = \mathbf{F}_q^l \left[\mathbf{p}_q^i\right] - \mathbf{F}_k^l \left[\mathbf{p}_k^i\right] \in \mathbb{R}^D \quad , \tag{3.1}$$

where $\mathbf{p}_q^i = \Pi\left(\mathbf{R}\mathbf{P}_i + \mathbf{t}\right)$ is the projection of i in the query given its current pose estimate and $[\cdot]$ is a lookup with sub-pixel interpolation. The total error over N

observations is

$$E_l(\mathbf{R}, \mathbf{t}) = \sum_{i,k} w_k^i \, \rho\left(\left\|\mathbf{r}_k^i\right\|_2^2\right) \quad , \tag{3.2}$$

where ρ is a robust cost function [113] with derivative ρ' and w_k^i is a per-residual weight. This nonlinear least-squares cost is iteratively minimized from an initial estimate $(\mathbf{R}_0, \mathbf{t}_0)$ using the Levenberg-Marquardt (LM) algorithm [167, 194].

To maximize the convergence basin, we optimize each feature level successively, starting with the coarsest level l=1, and initialize each with the result of the previous level. Low-resolution feature maps are thus responsible for the robustness of the pose prediction while finer features enhance its accuracy. Each pose update $\boldsymbol{\delta} \in \mathbb{R}^6$ is parametrized on the $\mathbf{SE}(3)$ manifold using its Lie algebra. We stack all residuals into $\mathbf{r} \in \mathbb{R}^{ND}$ and all weights into $\mathbf{W} = \mathrm{diag}_{i,k}\left(w_k^i \, \rho'\right)$ and write the Jacobian and Hessian matrices as

$$\mathbf{J}_{i,k} = \frac{\partial \mathbf{r}_k^i}{\partial \boldsymbol{\delta}} = \frac{\partial \mathbf{F}_q}{\partial \mathbf{p}_q^i} \frac{\partial \mathbf{p}_q^i}{\partial \boldsymbol{\delta}} \quad \text{and} \quad \mathbf{H} = \mathbf{J}^\top \mathbf{W} \mathbf{J} \ . \tag{3.3}$$

The update is computed by damping the Hessian and solving the linear system:

$$\boldsymbol{\delta} = -\left(\mathbf{H} + \lambda \, \mathrm{diag}\left(\mathbf{H}\right)\right)^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{r} \quad , \tag{3.4}$$

where λ, the damping factor, interpolates between the Gauss-Newton (λ=0) and gradient descent ($\lambda \to \infty$) formulations and is usually adjusted at each iteration using diverse heuristics [167, 192, 194]. Finally, the new pose is computed by left-multiplication on the manifold as

$$\begin{bmatrix} \mathbf{R}^+ & \mathbf{t}^+ \end{bmatrix} = \exp\left(\boldsymbol{\delta}^\wedge\right)^\top \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad , \tag{3.5}$$

where $\cdot^\wedge$ is the skew operator. The optimization stops when the update $\boldsymbol{\delta}$ is small enough.

Infusing visual priors: The steps described above are identical to the classical photometric alignment [18, 90, 184]. The CNN is however capable of learning complex visual priors – we therefore would like to give it the ability to steer the optimization towards the correct pose. To this end, the CNN predicts an uncertainty

map $\mathbf{U}_k^l \in \mathbb{R}_{>0}^{W_l \times H_l}$ along with each feature map. The pointwise uncertainties of the query and reference images are combined into a per-residual weight as

$$w_k^i = u_q^i \, u_k^i = \frac{1}{1 + \mathbf{U}_q^l \left[\mathbf{p}_q^i\right]} \frac{1}{1 + \mathbf{U}_k^l \left[\mathbf{p}_k^i\right]} \in [0, 1] \ . \tag{3.6}$$

The weight is 1 if the 3D point projects into a location with low uncertainty in both the query and the reference images. It tends to 0 as either of the location is uncertain. Here w_k^i is not explicitly supervised, but rather learned as to maximize the pose accuracy. A similar formulation was applied to direct RGB-D frame tracking in a concurrent work [357].

This weighting can capture multiple scenarios. First, the network can learn to be uncertain when it cannot predict invariant features, e.g., because of domain shift, similarly to an aleatoric uncertainty [148]. The uncertainty can also be high for locations that can be well described by the CNN, but which consistently push the optimization away from the correct pose by introducing local minima in the cost landscape. This encompasses dynamic objects or repeated patterns and symmetries, as shown in Figs. 3.4 and 3.7. The uncertainty is different for each level, as different cues might be useful at different stages of the optimization.

Fitting the optimizer to the data: Levenberg-Marquardt is a generic optimization algorithm that involves several heuristics, such as the choice of robust cost function ρ or of the damping factor λ. Past works on learned optimization employ deep networks to predict ρ' [186], λ [186, 311], or even the pose update $\boldsymbol{\delta}$ [71, 190], from the residuals and visual features. We argue that this can greatly impair the ability to generalize to new data distributions, as it ties the optimizer to the visual-semantic content of the training data. Instead, it is desirable to fit the optimizer to the distribution of poses or residuals but not to their semantic content. As such, we propose to make λ a fixed model parameter and learn it by gradient descent along with the CNN.

Importantly, we learn a different factor for each of the 6 pose parameters and for each feature level, replacing the scalar λ by $\boldsymbol{\lambda}_l \in \mathbb{R}^6$, parametrized by $\boldsymbol{\theta}_l$ as

$$\log_{10} \boldsymbol{\lambda}_l = \lambda_{\min} + \operatorname{sigmoid}\left(\boldsymbol{\theta}_l\right) \left(\lambda_{\max} - \lambda_{\min}\right) \ . \tag{3.7}$$

This adjusts the curvature of the individual pose parameters during training, and directly learns motion priors from the data. For example, when the camera is

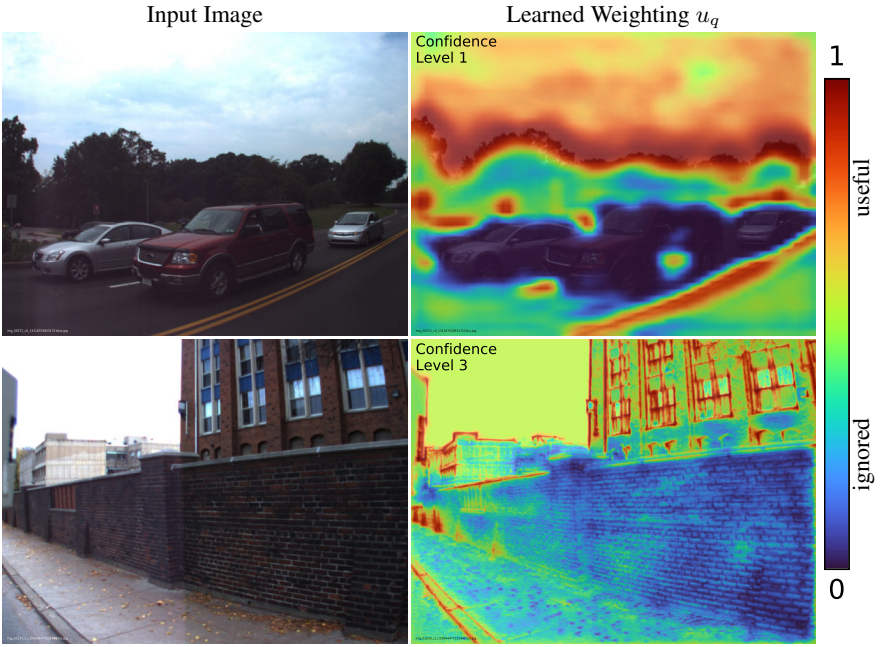


Figure 3.4.: ***Good features to localize.*** *PixLoc learns to ignore dynamic objects like cars (top) or fallen leaves (bottom) and repeated patterns like the brick wall. It focuses on road markings, silhouettes of trees, or prominent structures on buildings. See also Fig. 3.7.*

mounted on a car or a robot that is mostly upright, we expect the damping for the in-plane rotation to be large. In contrast, common heuristics treat all pose parameters equally and do not permit a per-parameter damping. We show in Sec. 3.7 that the learned damping parameters vary with the training data.

3.3.2. Learning from poses

As the CNN never sees 3D points, PixLoc can generalize to any 3D structure available. This includes sparse SfM point clouds, dense depth maps from stereo or RGBD sensors, meshes, Lidar scans, but also lines and other primitives.

Training: The optimization algorithm presented here is end-to-end differentiable and only involves operations commonly supported by deep learning frameworks. Gradients thus flow from the pose all the way to the pixels, through the feature and uncertainty maps and the CNN. Thanks to the uncertainties and robust cost, PixLoc is robust to incorrect 3D geometry and works well with noisy reference data like sparse SfM models. During training, an imperfect 3D representation is sufficient – our approach does not require accurate or dense 3D models.

Loss function: Our approach is trained by comparing the pose estimated at each level $(\mathbf{R}_l, \mathbf{t}_l)$ to its ground truth $(\bar{\mathbf{R}}, \bar{\mathbf{t}})$. We minimize the reprojection error of the 3D points:

$$\mathcal{L} = \frac{1}{L} \sum_l \sum_i \left\| \Pi\left(\mathbf{R}_l \mathbf{P}_i + \mathbf{t}_l\right) - \Pi\left(\bar{\mathbf{R}} \mathbf{P}_i + \bar{\mathbf{t}}\right) \right\|_\gamma , \tag{3.8}$$

where γ is the Huber cost. This loss weights the supervision of the rotation and translation adaptively for each training example [147] and is invariant to the scale of the scene, making it possible to train with data generated by SfM. To prevent hard examples from smoothing the fine features, we apply the loss at a given level only if the previous one succeeded in bringing the pose sufficiently close to the ground truth. Otherwise, the subsequent loss terms are ignored.

3.3.3. Comparisons to existing approaches

PixLoc vs. sparse matching: Pose estimation via local feature matching comprises multiple operations that are non-differentiable, such as keypoint and correspondence selection or RANSAC. Bhowmik et al. [32] proposed a formulation based on reinforcement learning, which suffers from high variance and thus requires a strong pretraining. In contrast, our approach is extremely simple and converges well from generic weights trained for image classification.

PixLoc vs. GN-Net: Von Stumberg et al. [339, 340] recently trained deep features for cross-season localization via direct alignment. Their works focus on small-baseline scenarios and require accurate pixelwise ground truth correspondences and substantial hyperparameter tuning. In contrast, we leverage the power of

Figure 3.5.: ***Wide convergence.*** *For a red point in the reference image (left), we highlight in the query (right) the multilevel basin of attraction colored by the 2D gradient angle* $\partial \mathbf{F}_q / \partial \mathbf{P}_q^i{}^\top \mathbf{r}_k^i$. *Deep features ensure a wide convergence despite appearance changes.*

differentiable programming to match the test and training conditions and learn additional strong priors from noisy data. We compare with their loss in Sec. 3.7.1.

3.4. Localization pipeline

PixLoc can be a competitive standalone localization module when coupled with image retrieval, but can also refine poses obtained by previous approaches. It only requires a 3D model and a coarse initial pose, which we now discuss.

Initialization: How accurate the initial pose should be depends on the basin of convergence of the alignment. Features from a deep CNN with a large receptive field ensure a large basin (Fig. 3.5). To further increase it, we apply PixLoc to image pyramids, starting at the lowest resolution, yielding coarsest feature maps of size $W{=}16$. To keep the pipeline simple, we select the initial pose as the pose of the first reference image returned by image retrieval. This results in a good convergence in most scenarios. When retrieval is not sufficiently robust and returns an incorrect location, as in the most challenging conditions, one could improve the performance by reranking using covisiblity clustering [261, 272] or pose verification with sparse [269, 368] or dense matching [308].

	Method	Cambridge Landmarks - outdoor					7Scenes - indoor							
		Court	King's	Hospital	Shop	St. Mary's	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Recall↑
IR	DenseVLAD [322]	-	280/5.7	401/7.1	111/7.6	231/8.0	21/12.5	33/13.8	15/14.9	28/11.2	31/11.3	30/12.3	25/15.8	-
	Oracle	207/7.0	137/7.2	323/8.3	133/7.8	204/8.1	16/12.3	26/13.6	12/14.7	20/11.5	19/14.0	18/15.0	17/18.1	0.17
FM	AS [272]†	24/0.13	13/0.22	20/0.36	4/0.21	8/0.25	3/0.87	2/1.01	1/0.82	4/1.15	7/1.69	5/1.72	4/1.01	68.7
	InLoc [308]	-	-	-	-	-	3/1.05	3/1.07	2/1.16	3/1.05	5/1.55	4/1.31	9/2.47	66.3
	hloc [261]	16/0.11	12/0.20	15/0.30	4/0.20	7/0.21	2/0.85	2/0.94	1/0.75	3/0.92	5/1.30	4/1.40	5/1.47	73.1
end-to-end	DSAC* [39]	49/0.3	15/0.3	21/0.4	**5**/0.3	13/0.4	**2**/1.10	**2**/1.24	**1**/1.82	**3**/1.15	**4**/1.34	4/1.68	**3**/1.16	**85.2**
	HACNet [168]	**28**/0.2	18/0.3	19/**0.3**	6/0.3	**9/0.3**	**2/0.7**	**2**/0.9	**1**/0.9	**3/0.8**	**4/1.0**	4/**1.2**	**3/0.8**	84.8
	CAMNet [83]	-	-	-	-	-	4/1.73	3/1.74	5/1.98	4/1.62	**4**/1.64	4/1.63	4/1.51	-
	SANet [360]	328/1.95	32/0.54	32/0.53	10/0.47	16/0.57	3/0.88	3/1.08	2/1.48	**3**/1.00	5/1.32	4/1.40	16/4.59	68.2
	PixLoc (SfM)	30/**0.14**	**14/0.24**	**16/0.32**	**5/0.23**	10/**0.34**	3/0.91	**2**/0.87	**1/0.77**	**3**/0.94	5/1.41	4/1.44	6/1.38	69.5
	+ oracle prior	21/0.12	13/0.24	16/0.31	5/0.22	9/0.28	3/0.86	2/0.89	1/0.76	3/0.95	5/1.40	4/1.42	6/1.32	71.0
	PixLoc (depth)	-	-	-	-	-	**2**/0.80	**2/0.73**	**1**/0.82	**3/0.82**	**4**/1.21	**3/1.20**	5/1.30	75.7
	+ oracle prior	-	-	-	-	-	2/0.80	2/0.70	1/0.78	3/0.80	4/1.13	3/1.14	4/1.08	81.7

Table 3.1.: ***Visual localization on the Cambridge Landmarks and 7Scenes datasets.*** *We report the median translation (cm) and rotation (°) errors and the average recall at (5cm, 5°). Despite its simplicity, PixLoc is competitive with complex feature matching (FM) pipelines and performs similarly to, and often better than, geometric regression algorithms, including those specifically trained per scene (red). Our approach, trained solely on outdoor data, generalizes well to unseen outdoor and indoor scenes. I can leverage 3D maps built with both SfM and depth sensors and can benefit from improved image retrieval (IR). The best results in the end-to-end category are in bold (oracle excluded). †The results for AS were kindly provided by the authors.*

3D structure: For simplicity and unless mentioned, for both training and evaluation, we use sparse SfM models triangulated from posed reference images using hloc [260, 261] and COLMAP [278, 281]. Given a subset of reference images, e.g. top-5 retrieved, we gather all the 3D points that they observe, extract multilevel features at their 2D observations, and average them based on their confidence.

3.5. Experiments

We first compare against existing learning-based localization approaches and show that PixLoc often performs better than those trained for each scene and generalizes well across environments. We then compare PixLoc with state-of-the-art feature matching pipelines on a large-scale benchmark and show that it delivers competitive accuracy, but can also enhance them when used as a post-processing. Finally, we provide insights into PixLoc through an ablation study.

Architecture: We employ a UNet feature extractor based on a VGG19 encoder pretrained on ImageNet, and extract L=3 feature maps with strides 1, 4, and 16, and dimensions D_l=32, 128, and 128, respectively. PixLoc is implemented in PyTorch [224], extracts features for an image in around 100ms, and optimizes the pose in 200ms to 1s depending on the number of points.

Training: We train two versions of PixLoc to demonstrate its ability to learn environment-specific priors. The benefits of such priors are analyzed in Sec. 3.7. One version is trained on the MegaDepth dataset [172], composed of crowd-sourced images depicting popular landmarks around the world, and the other on the training set of the Extended CMU Seasons dataset [14, 273, 318], a collection of sequences captured by car-mounted cameras in urban and rural environments. The latter dataset exhibits large seasonal changes with often only natural structures like trees being visible in the images, which are challenging for feature matching. We sample covisible image pairs and simulate the localization of one image with respect to the other, given its observed 3D points. The optimization runs for 15 iterations at each level and is initialized with the pose of the reference image.

	Method	Aachen Day-Night		RobotCar Seasons		Extended CMU Seasons		
		Day	Night	Day	Night	Urban	Suburban	Park
IR	DenseVLAD [322]	0.0 / 0.1 / 22.8	0.0 / 1.0 / 19.4	7.6 / 31.2 / 91.2	1.0 / 4.4 / 22.7	14.7 / 36.3 / 83.9	5.3 / 18.7 / 73.9	5.2 / 19.1 / 62.0
IR	NetVLAD [10]	0.0 / 0.2 / 18.9	0.0 / 0.0 / 14.3	6.4 / 26.3 / 90.9	0.3 / 2.3 / 15.9	12.2 / 31.5 / 89.8	3.7 / 13.9 / 74.7	2.6 / 10.4 / 55.9
IR	Oracle	0.0 / 0.2 / 22.1	0.0 / 1.0 / 22.4	9.6 / 38.1 / 96.3	4.3 / 16.4 / 84.9	21.2 / 52.2 / 98.2	8.6 / 29.5 / 94.3	8.2 / 31.5 / 90.2
E2E	ESAC [37]	42.6 / 59.6 / 75.5	6.1 / 10.2 / 18.4	-	-	-	-	-
E2E	**Pixloc**	64.3 / 69.3 / 77.4	51.0 / 55.1 / 67.3	52.7 / 77.5 / 93.9	12.0 / 20.7 / 45.4	88.3 / 90.4 / 93.7	79.6 / 81.1 / 85.2	61.0 / 62.5 / 69.4
E2E	+ Oracle prior	68.0 / 74.6 / 80.8	57.1 / 69.4 / 76.5	55.8 / 80.8 / 96.4	23.6 / 40.3 / 77.8	92.8 / 95.1 / 98.5	91.9 / 93.4 / 95.8	84.0 / 85.8 / 90.9
FM	AS [272]	85.3 / 92.2 / 97.9	39.8 / 49.0 / 64.3	50.9 / 80.2 / 96.6	6.9 / 15.6 / 31.7	81.0 / 87.3 / 92.4	62.6 / 70.9 / 81.0	45.5 / 51.6 / 62.0
FM	D2-Net [85]	84.3 / 91.9 / 96.2	75.5 / 87.8 / 95.9	54.5 / 80.0 / 95.3	20.4 / 40.1 / 55.0	94.0 / 97.7 / 99.1	93.0 / **95.7** / **98.3**	**89.2** / **93.2** / **95.0**
FM	S2DNet [105]	84.5 / 90.3 / 95.3	74.5 / 82.7 / 94.9	53.9 / 80.6 / 95.8	14.5 / 40.2 / 69.7	-	-	-
FM	hloc [261]	**89.6** / **95.4** / **98.8**	**86.7** / **93.9** / **100.**	**56.9** / 81.7 / **98.1**	33.3 / 65.9 / 88.8	95.5 / 98.6 / **99.3**	90.9 / 94.2 / 97.1	85.7 / 89.0 / 91.6
FM	**+ PixLoc refine**	84.7 / 94.2 / **98.8**	81.6 / **93.9** / **100.**	**56.9** / **82.0** / **98.1**	**34.9** / **67.7** / **89.5**	**96.9** / **98.9** / **99.3**	**93.3** / 95.4 / 97.1	87.0 / 89.5 / 91.6

Table 3.2.: ***Large-scale localization on the Aachen, RobotCar, and CMU datasets.*** *PixLoc, when initialized from image retrieval (IR), can substantially improve IR accuracy. It consistently outperforms the only scalable end-to-end (E2E) method ESAC, and performs reasonably compared to complex feature matching (FM) pipelines. PixLoc can also improve their accuracy by refining their local features (+ refine).*

3.5.1. Comparison to learned approaches

We first evaluate on the Cambridge Landmarks [149] and 7Scenes [291] datasets, which are commonly used to compare learning-based approaches.

Evaluation: The two datasets contain 5 outdoor and 7 indoor scenes, respectively, each composed of posed reference images and query images captured along different trajectories and conditions. We report for each scene the median translation (cm) and rotation (°) errors [149], as well as the average localization recall at (5cm, 5°) for 7Scenes [291].

Baselines: We compare with multiple state-of-the-art learning-based approaches. Those trained per scene include 3D coordinate regression networks DSAC* RGB [39] and HACNet [168], and CAMNet [83], which regresses a relative pose following image retrieval. SANet [360] is scene-agnostic. All approaches, including PixLoc, use 3D points from SfM and dense depth maps for Cambridge and 7Scenes, respectively. These depth maps were rendered by Brachmann et al. [39] from a mesh built by integrating the noisy depth sensor measurements and are aligned to the color images.

We report image retrieval with DenseVLAD [322] but not PoseNet and its variants as they perform similarly [275]. We also compare with feature matching pipelines. Active Search (AS) [272] performs global matching with SIFT [183]. InLoc [308] and hloc [261] first perform image retrieval before matching features to the retrieved images. The former matches dense deep descriptors and relies on a dense reference 3D model, while hloc matches SuperPoint [81] features with SuperGlue [263] and builds a sparse 3D SfM reference point cloud. PixLoc, trained on MegaDepth, is initialized with image retrieval obtained with either DenseVLAD [322] or an oracle, which returns the reference image containing the largest number of inlier matches found by hloc. This oracle shows the benefits of better image retrieval using a more complex pipeline without ground truth information.

Results: The evaluation results are reported in Tab. 3.1. On outdoor data, PixLoc consistently outperforms the only end-to-end scene-agnostic method, SANet, and performs similarly to, or better than scene-specific approaches. It is competitive for indoor scenes, despite being trained on outdoor Internet data only. This confirms

that deep features are all we need for accurate localization and that they generalize well despite end-to-end training. PixLoc performs comparably to the best feature matching localizer hloc – a complex pipeline that integrates learned feature detection, description, and matching. Localizing with the oracle prior only marginally improves the performance, confirming that image retrieval can be sufficiently accurate for the pose optimization to converge to the correct minimum. On 7Scenes, using dense depth maps yields slightly more accurate poses than using the sparse SfM point cloud.

3.5.2. Large-scale localization

We now evaluate on a large-scale, long-term localization benchmark [273] that exhibits considerably more diversity in geometry and appearance than Cambridge and 7Scenes.

Evaluation: The benchmark is composed of three datasets. The Aachen Day-Night [273, 274] dataset is captured by handheld devices. The RobotCar [191, 273] and the Extended CMU [14, 318] seasons datasets are captured by car-mounted cameras across different seasons, weather, and times, in urban and rural areas. All datasets have posed reference images, SfM models, and query images. We report the localization recall at thresholds (25cm, 2°), (50cm, 5°), and (5m, 10°).

Baselines: Multiple past works [37, 275, 280, 308] report that end-to-end learning-based methods cannot be stably trained on such large-scale datasets. The only exception is ESAC [37], which reports results for Aachen only. We additionally compare against image retrieval with DenseVLAD [322] and NetVLAD [10] and feature matching pipelines based on Active Search [272], D2-Net [85], S2DNet [105], and hloc [261]. PixLoc is trained on MegaDepth (CMU) when evaluated on Aachen (RobotCar and CMU). It is initialized by the weighted average [233] of the top-3 poses retrieved by NetVLAD for Aachen and top-1 for RobotCar and CMU. The oracle prior is identical to Sec. 3.5.1.

Results: We report the results in Tab. 3.2. When the initial pose prior is provided by image retrieval, PixLoc is a simple localization system that is more accurate than ESAC, especially in the challenging condition of night. This improvement is not

brought by the significantly less accurate image retrieval. PixLoc is however less robust than the feature matching pipelines, which is mostly due to the naive pose prior, as our algorithm cannot converge if the retrieval returns the incorrect location. Using the oracle prior partially bridges the gap, and makes PixLoc competitive on driving datasets like CMU and RobotCar. It however lags behind on Aachen, where the reference images are significantly sparser and the initial priors are therefore much coarser. Naturally, this is challenging for direct alignment, irrespective of the daytime or nighttime condition. PixLoc is nevertheless the only end-to-end trained method that can scale to this large extent without requiring retraining.

3.5.3. Pose post-processing with PixLoc

We showed that too large baselines between query and reference images can cause PixLoc to converge to an incorrect local minima. Naturally, PixLoc can also serve as a post-processing step for any other localization pipeline.

Refinement in challenging conditions: We apply PixLoc to refine the poses estimated by hloc in the previous localization experiment. We consider all 3D points that have at least one inlier match. The results are shown in the last row of Tab. 3.2. PixLoc brings consistent improvement on CMU, especially in the fine threshold, with up to +2.4% recall. It also increases the pose accuracy at all thresholds on RobotCar Night, which exhibits significant motion blur, a difficult condition for sparse keypoint detection. However, no improvement can be observed on RobotCar Day, while the refinement is detrimental on Aachen at 0.25m. This might be due to inaccurate ground truth poses or camera intrinsics.

3.6. Convergence and initial pose

Convergence: The pose optimization in PixLoc tends to converge to spurious local minima if the initial pose is too coarse, such as on the Aachen dataset, in which reference images are sparse. Since the receptive field of the CNN is limited, the convergence mostly depends on the initial 2D reprojection error, which accounts for the rotation and translation errors and for the distance to the 3D structure. The exact

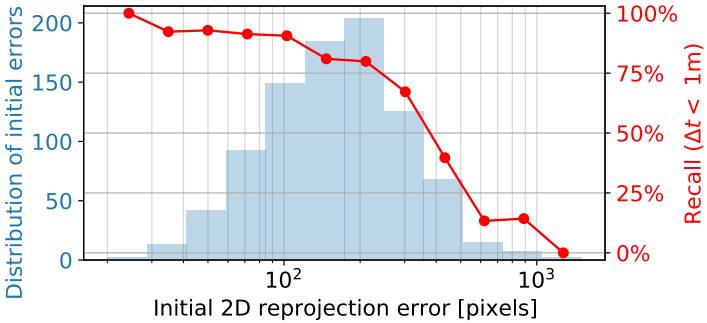


Figure 3.6.: ***Impact of the initial pose on the Aachen dataset.*** *The success of the pose optimization decreases with larger initial reprojection errors, which vary significantly across the 922 queries.*

Initial pose	Aachen Day-Night		CMU Seasons
	Day	Night	Park
top-1	61.7 / 67.6 / 74.8	46.9 / 53.1 / 64.3	61.0 / 62.5 / 69.4
top-3 averaging	64.3 / 69.3 / 77.4	51.0 / 55.1 / 67.3	64.9 / 66.8 / 71.7
oracle prior	68.0 / 74.6 / 80.8	57.1 / 69.4 / 76.5	84.0 / 85.8 / 90.9

Table 3.3.: ***Selection of the initial pose.*** *Averaging the poses of the top retrieved images improves the convergence of PixLoc compared to simply selecting the pose of the first image.*

density of reference images required for high success thus depends on the distance to the scene.

We report in Fig. 3.6 the success rate for different initial reprojection errors and their distribution for the oracle retrieval, with hloc as pseudo ground truth. Convergence within 1 meter is observed for 80% of the cases only when the initial error is smaller than 200 pixels and is significantly reduced for larger errors.

Initial pose: The 7Scenes and Cambridge datasets have reference poses with a high density. In driving scenarios like in the RobotCar and CMU datasets, there are no rotation changes between reference and query poses. In all these scenarios, initializing PixLoc with the pose of the first retrieved image is therefore sufficient.

To improve the performance on the Aachen dataset, the results in Tab. 3.2 rely

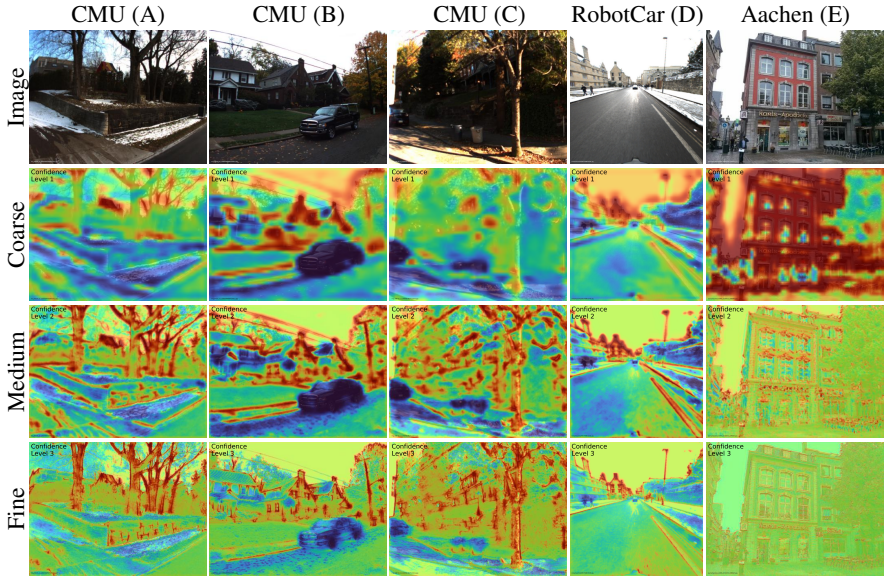


Figure 3.7.: ***Which features do matter?*** *In driving scenarios (A-D), besides dynamic objects such as cars, PixLoc learns to ignore (in blue) more subtle short-term entities like snow (A), fallen leaves (B), trash bins (C), or shadows at all feature levels. Instead, it focuses (in red) on poles, tree trunks, road markings, power lines, or building silhouettes. Repetitive structures like windows or road cracks are often ignored at first but later on used for fine alignment. Differently, when trained on urban scenes (E), it ignores trees as buildings are more stable structures.*

on additional filtering steps. We first cluster the top-3 retrieved reference images based on their covisibility [261, 272] and only retain the images that belong to the largest cluster. We then perform a weighted average of the reference poses [193], where the weights are computed from the similarity of the global descriptors [233]. We compare in Tab. 3.3 the results obtained with this pose averaging and with the top-1 retrieval. To further improve the convergence, one could also rerank based on featuremetric error or initialize with poses randomly sampled around top-retrieved poses.

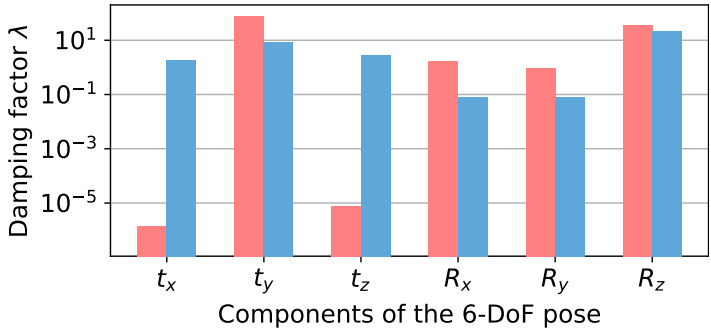


Figure 3.8.: ***Learned motion prior.*** *Training on data recorded with 3-DoF car-mounted cameras (CMU, in red) or with 6-DoF hand-held devices (MegaDepth, in blue) results in different motion priors learned by the damping factor* $\boldsymbol{\lambda}$*. Larger relative values indicate smaller expected motion in the corresponding direction.*

Training dataset	Aachen (urban scenes like MD)		CMU (natural scenes)	
	Day	Night	Urban	Park
MD	**68.0 / 74.6 / 80.8**	**57.1 / 69.4 / 76.5**	78.3 / 81.8 / 94.6	72.5 / 75.5 / 90.3
CMU	54.4 / 62.6 / 74.3	46.9 / 54.1 / 68.4	**91.9 / 93.4 / 95.8**	**84.0 / 85.8 / 90.9**

Table 3.4.: ***Cross-dataset evaluation with oracle prior.*** *Training and testing in different environments does not perform as well as training for the target distribution. Task-specific priors learned by PixLoc, like semantics and motion, are thus largely beneficial.*

3.7. Benefits of training on different datasets

Semantic priors: The training datasets CMU and MegaDepth reflect different scenarios, autonomous driving and tourism landmark photography, respectively. Training on each one separately allows to learn task-specific priors and demonstrates the ability of PixLoc to adapt to the environment. Each dataset depicts scenes with different semantic elements (street-level landscapes and urban landmarks, respectively) and different changes of conditions (weather and season for CMU, cameras, occluders, and viewpoints for MegaDepth). Fig. 3.7 shows that the models learn to ignore different unreliable elements depending on the training dataset. For

Method	AUC	
	25cm	1m
● Photometric optimization	1.2	3.3
● + deep features (GN loss)	13.2	21.4
● + unroll (fixed damping λ)	49.2	67.0
● + confidence $w_{i,k}$	53.3	72.5
● + learned λ **(PixLoc-full)**	**59.8**	**79.0**
● D=128→16 (PixLoc-light)	50.4	70.2

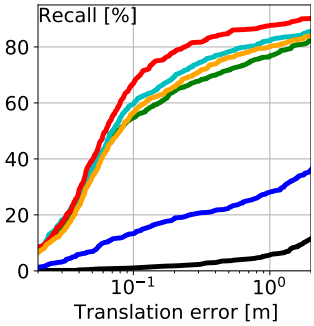


Table 3.5.: ***Ablation study.*** *Unrolling the optimizer and learning features, damping factor, and confidences all contribute to the performance of PixLoc over classical photometric alignment. Learning compact features as in past works [186, 339] results in a drop of performance compared to high-dimensional representations.*

example, tree silhouettes are reliable on CMU due to the small viewpoint changes, but are ignored by the model trained on MegaDepth.

Motion priors: Cameras also exhibit different motions, as they are either car-mounted or hand-held. Such priors are learned by the model through the damping factors, which we visualize in Fig. 3.8. On CMU, the motion across query and reference images is mostly a translation along the x and z axis of the camera, and never along the y axis (fixed height above the ground plane) or a rotation around the z axis (fixed roll). Differently, the motion on MegaDepth is more uniformly distributed among the 6 DoF, resulting in similar factors. The relative scale between the two sets of factors is irrelevant.

Evaluation: These learned priors have a noticeable impact on the performance, as shown in Tab. 3.4. Training on CMU performs better than training on MegaDepth when evaluating on a driving dataset like RobotCar. When evaluating on a totally different environment like Aachen, it however still performs better than a scene-specific approach like ESAC (shown in Tab. 3.2). PixLoc thus generalizes well across scenarios but can also learn and exploit their specificities.

3.7.1. Additional insights

Ablation study: We justify our design decisions by comparing different variants of PixLoc. We have attempted to train our CNN with the Gauss-Newton (GN) loss [339], but it fails to converge on our challenging training data despite extensive hyperparameter tuning. We select difficult query-reference pairs in the CMU validation set and report the recall curve and its area (AUC) in Tab. 3.5. As can be seen, all components significantly contribute to PixLoc's performance.

Interpretability: Visualizing the weight maps u_q learned by PixLoc helps us discover what cues are useful or detrimental for localizing in which environments (Fig. 3.7). We show examples of successful and failed localization in Figs. 3.9 and 3.11 and Figs. 3.10 and 3.12, respectively.

3.8. Summary and outlook

Summary: In this chapter, we have introduced a simple solution to end-to-end learning of camera pose estimation. In contrast to previous approaches that regress geometric quantities, we do not try to teach a deep network basic geometric principles or 3D map encoding. Instead, we go Back to the Feature: we show that learning robust and generic features is sufficient for accurate localization by leveraging classical image alignment with existing 3D maps. To the best of our knowledge, the resulting system, PixLoc, is the first end-to-end trainable approach capable of being deployed into new scenes widely differing from its training data without retraining or fine-tuning. PixLoc achieves a pose accuracy competitive with significantly more complex state-of-the-art pipelines. End-to-end training combined with uncertainty modeling enables PixLoc to learn complex yet interpretable priors.

PixLoc learns which features and objects matter for robust, long-term localization. Yet, it requires a good initialization to successfully localize. We thus see PixLoc as a first step towards deep networks that learn and reason about long-term, extreme changes of appearance and 3D structure. We believe that taking steps towards human-level spatiotemporal understanding will ultimately lead to robust, reliable, and accurate localization systems.

Limitations: PixLoc relies on gradients of CNN features, which can only encode a limited context. It is thus a local method and can fall into incorrect minima for excessively large initial reprojection errors arising from large viewpoint changes. Our approach is thus limited in terms of the viewpoint changes that it can handle. PixLoc can also fail for large outliers ratios due prominent occluders and is more sensitive to camera miscalibration.

More critically, we found that featuremetric errors are much less reliable to flag failure cases than the inlier count typically estimated by RANSAC. We hypothesize that, because they depend on the underlying image content, the featuremetric errors learned by PixLoc cannot effectively distinguish correct and incorrect alignments but only provide a local direction of minimization. Training such features to provide an absolute scoring, such as in a RANSAC scheme, could improve their reliability – but no experiment has confirmed this so far. Additionally, the optimization used in PixLoc is rather slow when accounting for all levels and scales. These are significant obstacles to the integration of PixLoc in a real-world system.

Impact: Given these limitations, PixLoc has had a relatively limited practical impact. The refinement introduced in Chapter 2 is more suitable for most scenarios. PixLoc has however inspired several lines of work that cast the estimation of a relative pose between multiple sensor modalities as a feature alignment process. Fu et al. [100] show that this can be used to refine the extrinsic calibration of cameras and LiDAR measurements. Shi et al. [286] localize a ground-level image within a satellite image. Our differentiable LM optimization is a critical component of both approaches. Veicht et al. [337] repurpose this optimization for the task of single-image camera calibration. PixLoc has also inspired other works that successfully integrate this optimization into larger systems [314, 315].

As for the task of pose refinement, Solonets et al. [296] extend the basin of convergence by explicitly controlling and annealing the smoothness of the features instead of learning a feature pyramid. Trivigno et al. [325] show that generic features are effective when a 3D model of the scene is available for rendering.

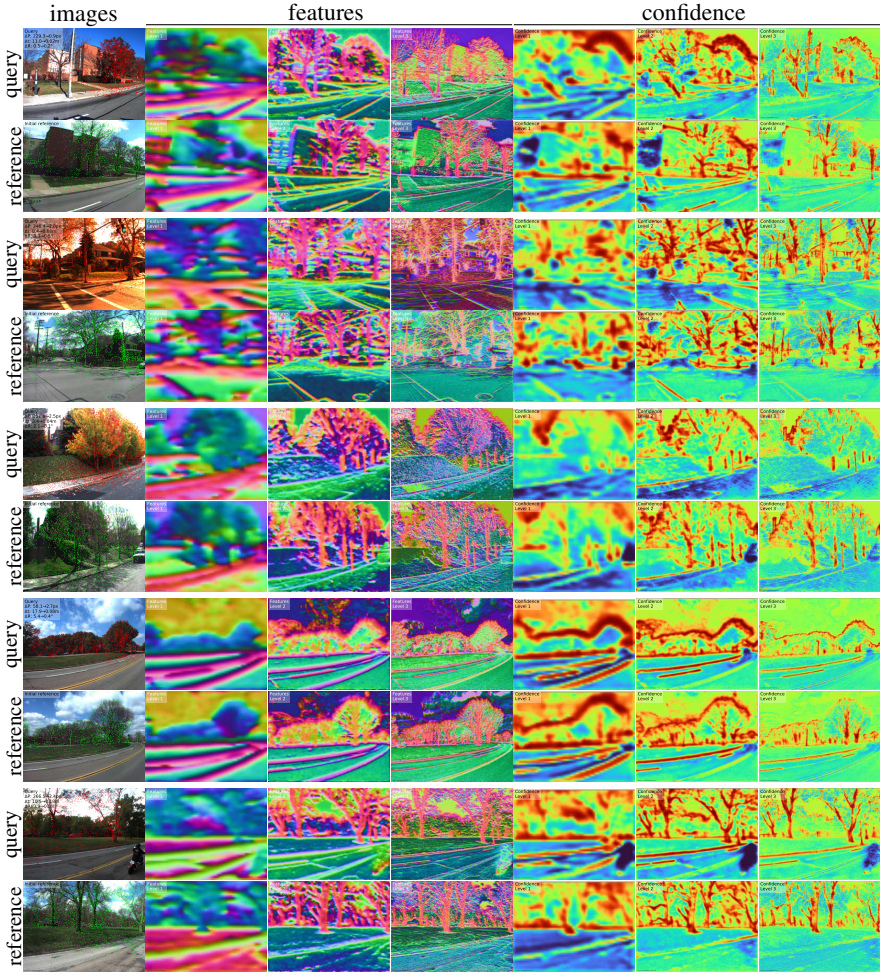


Figure 3.9.: ***Successful localization on the CMU dataset.*** *We show 5 challenging queries with large initial errors and large cross-season appearance changes that are successfully localized by PixLoc. We project 3D SfM points into the initial reference image (in green) and into the query image using the estimated pose (in red). We show the features at the 3 different levels, mapping them to RGB using PCA. We also show the confidence maps, where blue pixels are ignored while red ones are more important for the optimization. Features useful for localization are invariant across seasons and thus appear in similar colors.*

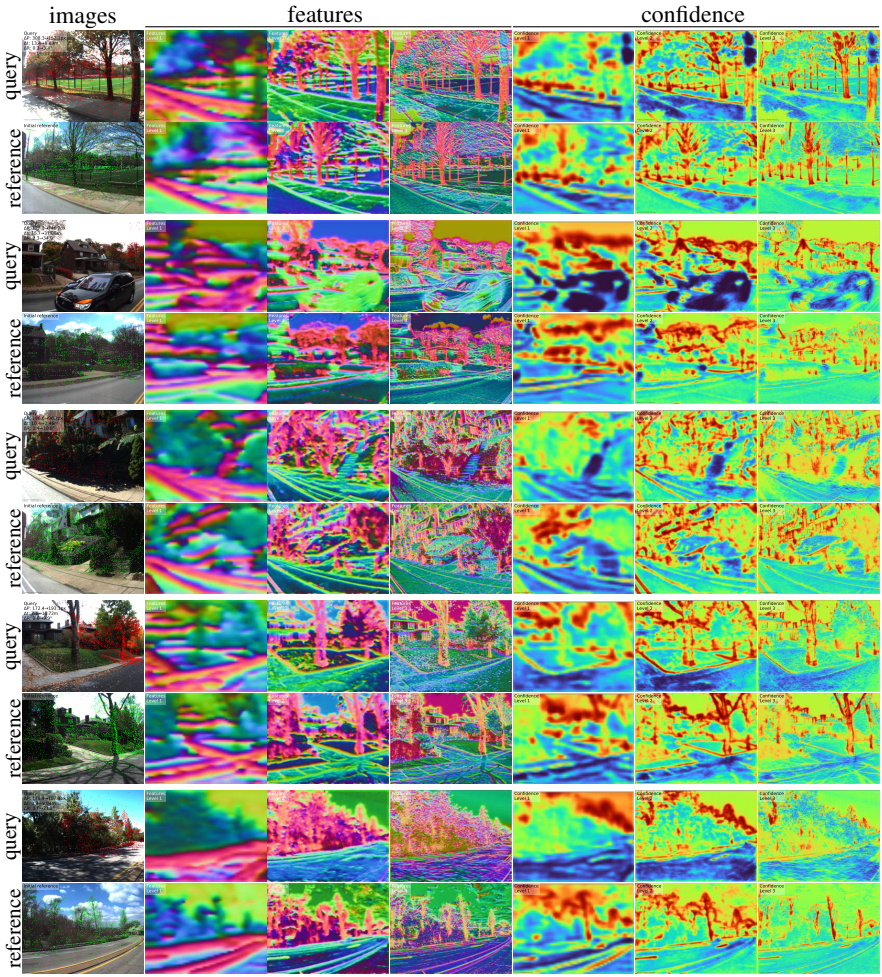


Figure 3.10.: ***Failure cases on the CMU dataset.*** *We show examples for which the optimization results in a large final error. This is often due to repeated elements or to a lack of spatial context of the coarse features or a lack of distinctive elements. Natural scenes are particularly challenging when tree trunks and vegetation cannot be easily distinguished.*

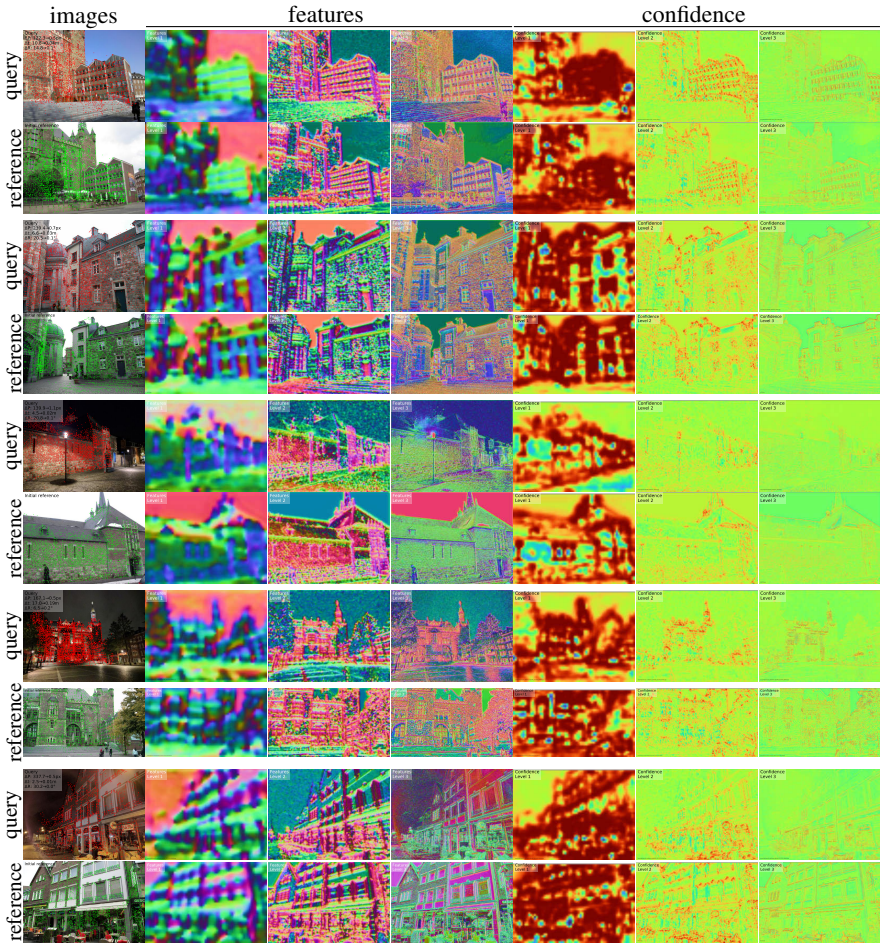


Figure 3.11.: ***Successful localization on the Aachen dataset.*** *We show 5 challenging queries with large initial errors and large day-night appearance changes that are successfully localized by PixLoc. The reprojection and pose errors are computed with respect to the pose estimated by hloc.*

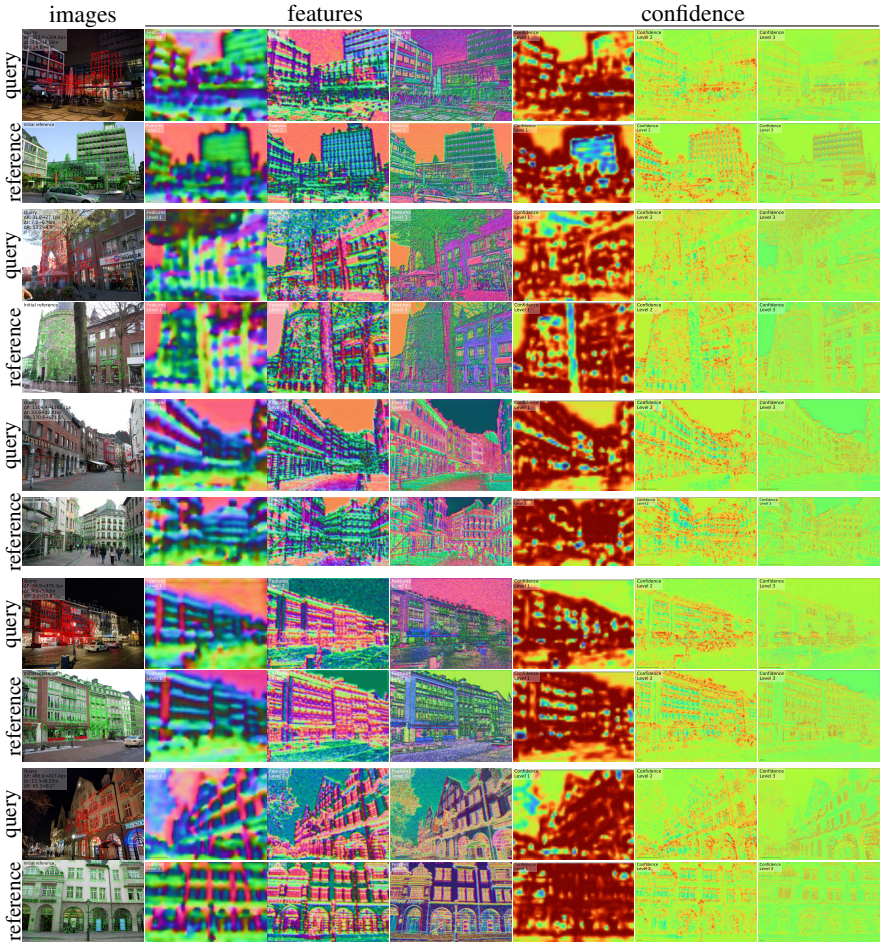


Figure 3.12.: ***Failure cases on the Aachen dataset.*** *Convergence to a local and incorrect minima can be due to large appearance changes (row 1), occlusion (row 2), large viewpoint change (row 3) or repeated structures on facades (rows 4 and 5).*

CHAPTER 4

Benchmarking Localization and Mapping for Augmented Reality

Localization and mapping is the foundational technology for Augmented Reality (AR) that enables sharing and persistence of digital content in the real world. While significant progress has been made, researchers are still mostly driven by unrealistic benchmarks not representative of real-world AR scenarios. These benchmarks are often based on small-scale datasets with low scene diversity, captured from stationary cameras, and lack other sensor inputs like inertial, radio, or depth data. Furthermore, their ground truth (GT) accuracy is mostly insufficient to satisfy AR requirements. To close this gap, we introduce LaMAR, a new benchmark with a comprehensive capture and GT pipeline that co-registers realistic trajectories and sensor streams captured by heterogeneous AR devices in large, unconstrained scenes. To establish an accurate GT, our pipeline robustly aligns the trajectories against laser scans in a fully automated manner. As a result, we publish a benchmark dataset of diverse and large-scale scenes recorded with head-mounted and hand-held AR devices. We extend several state-of-the-art methods to take advantage of the AR-specific setup and evaluate them on our benchmark. The results offer new insights on current research and reveal promising avenues for future work in the field of localization and mapping for AR.

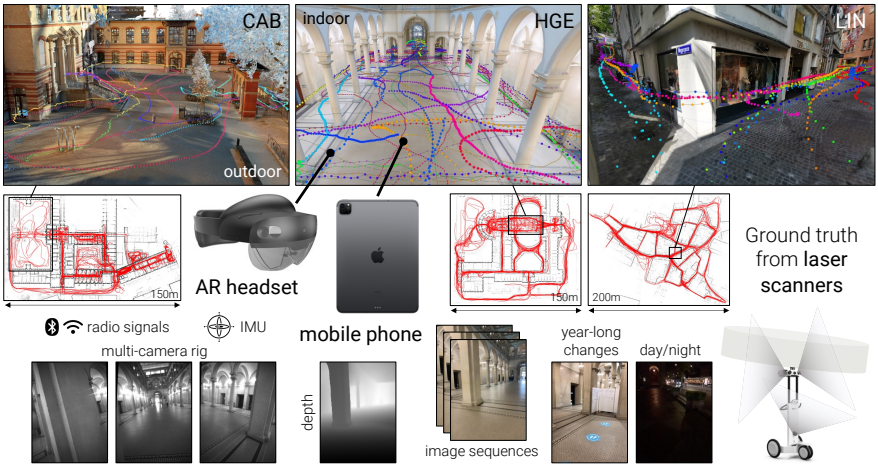


Figure 4.1.: *We revisit localization and mapping in the context of Augmented Reality by introducing LaMAR, a large-scale dataset captured using AR devices (HoloLens 2, iPhone) and laser scanners.*

4.1. Introduction

Placing virtual content in the physical 3D world, persisting it over time, and sharing it with other users are typical scenarios for AR. In order to reliably overlay virtual content in the real world with pixel-level precision, these scenarios require AR devices to accurately determine their 6-DoF pose at any point in time. While visual localization and mapping is one of the most studied problems in computer vision, its use for AR entails specific challenges and opportunities. First, modern AR devices, such as mobile phones or the Microsoft HoloLens or MagicLeap One, are often equipped with multiple cameras and additional inertial or radio sensors. Second, they exhibit characteristic hand-held or head-mounted motion patterns. The on-device real-time tracking systems provide spatially-posed sensor streams. However, many AR scenarios require positioning beyond local tracking, both indoors and outdoors, and robustness to common temporal changes of appearance and structure. Furthermore, given the plurality of temporal sensor data, the question is often not whether, but how quickly can the device localize at any time to ensure a compelling end-user experience. Finally, as AR adoption grows, crowd-sourced data captured

by users with diverse devices can be mined for building large-scale maps without a manual and costly scanning effort. Crowd-sourcing offers great opportunities but poses additional challenges on the robustness of algorithms, e.g., to enable cross-device localization [84], mapping from incomplete data with low accuracy [34, 278], privacy-preservation of data [87, 103, 104, 290, 299], etc.

However, the academic community is mainly driven by benchmarks that are disconnected from the specifics of AR. They mostly evaluate localization and mapping using single still images and either lack temporal changes [243, 291] or accurate GT [149, 273, 308], are restricted to small scenes [19, 149, 283, 291, 343] or landmarks [143, 279] with perfect coverage and limited viewpoint variability, or disregard temporal tracking data or additional visual, inertial, or radio sensors [49, 166, 273, 274, 305, 308].

Our first contribution is to introduce **a large-scale dataset captured using AR devices in diverse environments**, notably a historical building, a multi-story office building, and part of a city center. The initial data release contains both indoor and outdoor images with illumination and semantic changes as well as dynamic objects. Specifically, we collected multi-sensor data streams (images, depth, tracking, inertial measurements, Bluetooth, WiFi) totaling more than 100 hours using head-mounted HoloLens 2 and hand-held iPhone / iPad devices covering 45'000 square meters over the span of one year (Fig. 4.1).

Second, we develop **a GT pipeline to automatically and accurately register AR trajectories** against large-scale 3D laser scans. Our pipeline does not require any manual labeling or setup of custom infrastructure (e.g., fiducial markers). Furthermore, the system robustly handles crowd-sourced data from heterogeneous devices captured over longer periods of time and can be easily extended to support future devices.

Finally, we present **a rigorous evaluation of localization and mapping in the context of AR** and provide **novel insights for future research**. Notably, we show that the performance of state-of-the-art methods can be drastically improved by considering additional data streams generally available in AR devices, such as radio signals or sequence odometry. Thus, future algorithms in the field of AR localization and mapping should always consider these sensors in their evaluation to show real-world impact.

The LaMAR dataset, benchmark, GT pipeline, and the implementations of baselines integrating additional sensory data are all publicly available at `lamar.ethz.ch`. We hope that this will spark future research addressing the challenges of AR.

4.2. Related work

Image-based localization is classically tackled by estimating a camera pose from correspondences established between sparse local features [29, 183, 199, 253] and a 3D Structure-from-Motion (SfM) [278] map of the scene [95, 171, 271]. This pipeline scales to large scenes using image retrieval [11, 47, 139, 241, 244, 321, 322]. Recently, many of these steps or even the end-to-end pipeline have been successfully learned with neural networks [10, 81, 85, 136, 176, 203, 261, 263, 268, 280, 317, 364]. Other approaches regress absolute camera pose [147, 149, 211] or scene coordinates [37, 39, 169, 196, 197, 291, 335, 350]. However, all these approaches typically fail whenever there is lack of context (e.g., limited field-of-view) or the map has repetitive elements. Leveraging the sequential ordering of video frames [144, 201] or modelling the problem as a generalized camera [122, 235, 273, 299] can improve results.

Radio-based localization: Radio signals, such as WiFi and Bluetooth, are spatially bounded (logarithmic decay) [17, 120, 151], thus can distinguish similarly looking (spatially distant) locations. Their unique identifiers can be uniquely hashed which makes them computationally attractive (compared with high-dimensional image descriptors). Several methods use the signal strength, angle, direction, or time of arrival [56, 73, 227] but the most popular is model-free map-based fingerprinting [120, 151, 160], as it only requires to collect unique identifiers of nearby radio sources and received signal strength. GNSS provides absolute 3-DoF positioning but is not applicable indoors and has insufficient accuracy for AR scenarios, especially in urban environments due to multi-pathing, etc.

Datasets and ground-truth: As shown in Tab. 4.1, many of the existing benchmarks are captured in small-scale environments [76, 126, 291, 343], do not contain sequential data [19, 60, 143, 274, 279, 283, 305, 308], lack characteristic handheld/head-mounted motion patterns [15, 191, 273, 352], or their GT is not accurate enough for AR [149, 243]. None of these datasets contain WiFi or Bluetooth data.

dataset	out/indoor	changes	scale	density	camera motion	imaging devices	additional sensors	ground truth	accuracy
Aachen [273, 274]	✓ ✗	🌓 🏗	★★☆	★★☆	still images	DSLR	✗	SfM	>dm
Phototourism [143]	✓ ✗	🚶 🏗	☆☆☆	★★★	still images	DSLR, phone	✗	SfM	∼m
San Francisco [60]	✓ ✗	🚶 🏗	★★★	★☆☆	still images	DSLR, phone	GNSS	SfM+GNSS	∼m
Cambridge [149]	✓ ✗	🚶 🌧	☆☆☆	★★☆	handheld	mobile	✗	SfM	>dm
7Scenes [291]	✗ ✓	✗	☆☆☆	★★★	handheld	mobile	depth	RGB-D	∼cm
RIO10 [343]	✗ ✓	🪑	☆☆☆	★★★	handheld	Tango tablet	depth	VIO	>dm
InLoc [308]	✗ ✓	🪑	★☆☆	☆☆☆	still images	panoramas, phone	lidar	manual+lidar	>dm
Baidu mall [305]	✗ ✓	🚶	★☆☆	★★☆	still images	DSLR, phone	lidar	manual+lidar	∼dm
Naver Labs [166]	✗ ✓	🚶 🪑	★★☆	★★☆	robot-mounted	fisheye, phone	lidar	lidar+SfM	∼dm
NCLT [49]	✓ ✓	🌧 🪑	★★☆	★★☆	robot-mounted	wide-angle	lidar, IMU, GNSS	lidar+VIO	∼dm
ADVIO [243]	✓ ✓	🚶	★★☆	☆☆☆	handheld	phone, Tango	IMU, depth, GNSS	manual+VIO	∼m
ETH3D [283]	✓ ✓	✗	☆☆☆	★★☆	handheld	DSLR, wide-angle	lidar	manual+lidar	∼mm
LaMAR (ours)	✓ ✓	🚶 🌧 🌓 🪑 🏗	★★☆ 3 locations 45'000 m^2	★★★ 100 hours 40 km	handheld head-mounted	phone, headset backpack, trolley	lidar, IMU, 📶 🔵 depth, infrared	lidar+SfM+VIO automated	∼cm

Table 4.1.: ***Overview of existing datasets.*** *No dataset, besides ours, exhibits at the same time short-term appearance and structural changes due to moving people 🚶, weather 🌧, or day-night cycles 🌓, but also long-term changes due to displaced furniture 🪑 or construction work 🏗.*

device	motion type	cameras					radios	other data	poses
		#	FOV	freq.	resolution	specs			
M6	trolley	6	113°	1-3m	1080p	RGB, sync	📶 🔵	lidar points+mesh	lidar SLAM
VLX	backpack	4	90°	1-3m	1080p	RGB, sync	🔵	lidar points+mesh	lidar SLAM
HoloLens2	head-mounted	4	83°	30Hz	VGA	gray, GS	📶 🔵	ToF depth/IR 1Hz, IMU	head-tracking
iPad/iPhone	hand-held	1	64°	10Hz	1080p	RGB, RS, AF	🔵*	lidar depth 10Hz, IMU	ARKit

Table 4.2.: ***Sensor specifications.*** *Our dataset has visible light images (global shutter GS, rolling shutter RS, auto-focus AF), depth data (ToF, LiDAR), radio signals (*, if partial), dense LiDAR point clouds, and poses with intrinsics from on-device tracking.*

The closest to our work are Naver Labs [166], NCLT [49] and ETH3D [283]. Both, Naver Labs [166] and NCLT [49] are less accurate than ours and do not contain AR specific trajectories or radio data. The Naver Labs dataset [166] also does not contain any outdoor data. ETH3D [283] is highly accurate, however, it is only small-scale, does not contain significant changes, or any radio data.

To establish ground-truth, many datasets rely on off-the-shelf SfM algorithms [278] for unordered image collections [143, 143, 149, 243, 274, 305, 308, 343]. Pure SfM-based GT generation has limited accuracy [34] and completeness, which biases the evaluations to scenarios in which visual localization already works well. Other approaches rely on RGB(-D) tracking [291,343], which usually drifts in larger scenes and cannot produce GT in crowd-sourced, multi-device scenarios. Specialized capture rigs of an AR device with a more accurate sensor (LiDAR) [49, 166] prevent capturing of realistic AR motion patterns. Furthermore, scalability is limited for these approaches, especially if they rely on manual selection of reference images [305], laborious labeling of correspondences [274, 308], or placement of fiducial markers [126]. For example, the accuracy of ETH3D [283] is achieved by using single stationary LiDAR scan, manual cleaning, and aligning very few images captured by tripod-mounted DSLR cameras. Images thus obtained are not representative for AR devices and the process cannot scale or take advantage of crowd-sourced data. In contrast, our fully automatic approach does not require any manual labeling or special capture setups, thus enables light-weight and repeated scanning of large locations.

4.3. Dataset

We first give an overview of the setup and content of our dataset.

Locations: The initial release of the dataset contains 3 large locations representative of AR use cases: 1) HGE (18'000 m^2) is the ground floor of a historical university building composed of multiple large halls and large esplanades on both sides. 2) CAB (12'000 m^2) is a multi-floor office building composed of multiple small and large offices, a kitchen, storage rooms, and 2 courtyards. 3) LIN (15'000 m^2) is a few blocks of an old town with shops, restaurants, and narrow passages. HGE and CAB contain both indoor and outdoor sections with many symmetric structures.

Figure 4.2.: ***The CAB location*** *features 1-2) a staircase spanning 5 similar-looking floors, 3) large and small offices and meeting rooms, 4) long corridors, 5) large halls, and 6) outdoor areas with repeated structures. This location includes the* Facade, Courtyard, Lounge, Old Computer, Storage Room, *and* Office *scenes of the ETH3D [283] dataset.*

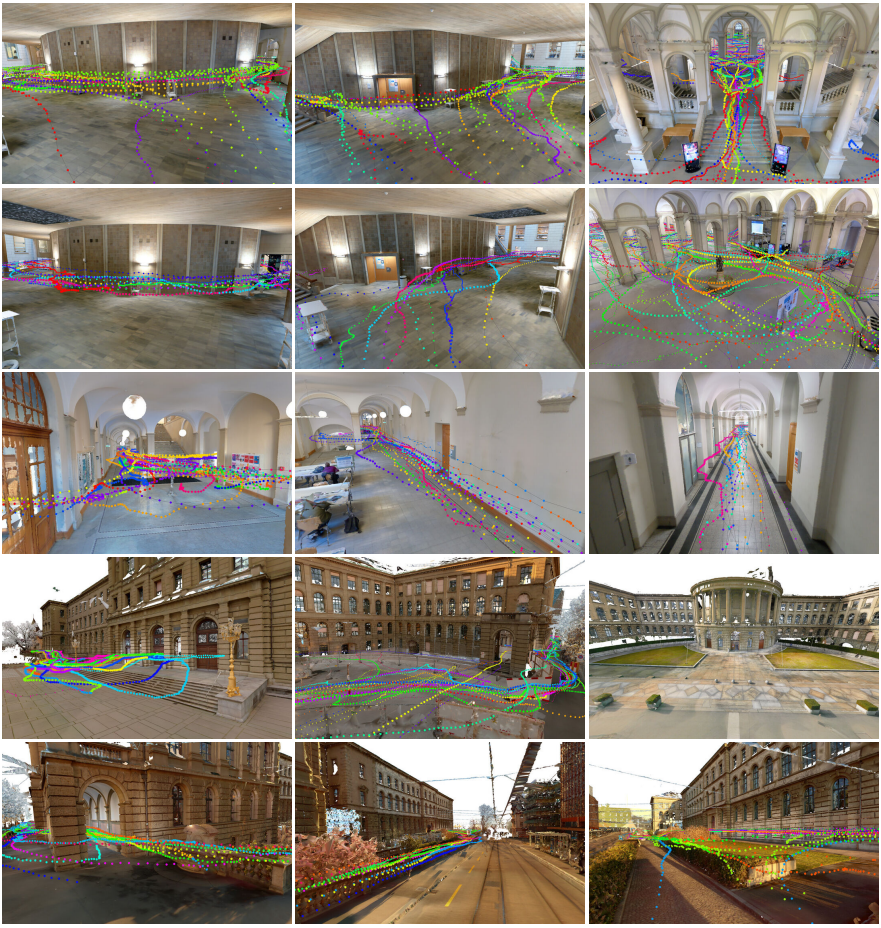


Figure 4.3.: ***The HGE location*** *features a highly-symmetric building with 1-2) hallways, 3) long corridors, 4) two esplanades, and 5) a section of sidewalk. This location includes the* Relief, Door, *and* Statue *scenes of the ETH3D [283] dataset.*

Figure 4.4.: ***The LIN location*** *features large outdoor open spaces (top row), narrow passages with stairs (middle row), and both residential and commercial street-level facades.*

Data collection: We collected data using Microsoft HoloLens 2 and Apple iPad Pro devices with custom raw sensor recording applications. 10 participants were each given one device and asked to walk through a common designated area. They were only given the instructions to freely walk through the environment to visit, inspect, and find their way around. This yielded diverse camera heights and motion patterns. Their trajectories were not planned or restricted in any way. Participants visited each location, both during the day and at night, at different points in time over the course of up to 1 year. In total, each location is covered by more than 100 sessions of 5 minutes. We did not need to prepare the capturing site in any way before recording. This enables easy barrier-free crowd-sourced data collections. Each location was also captured two to three times by NavVis M6 trolley or VLX backpack mapping platforms, which generate textured dense 3D models of the environment using laser scanners and panoramic cameras.

We show renderings of the resulting high-quality meshes along with trajectories of the numerous AR sequences, each shown as a different color, for each location in

in Figs. 4.2 to 4.4. Because spaces are actively used and managed, they undergo significant appearance and structural changes over the year-long data recording. For example, the front of the HGE building turned into a construction site and the indoor furniture was rearranged. This is captured by the laser scans, which are aligned based on elements that do not change, such as the structure of the buildings. We show in Fig. 4.5 a visual comparison between scans captured at different points in time.

Privacy: We paid special attention to comply with privacy regulations. Since the dataset is recorded in public spaces, our pipeline anonymizes all visible faces and licence plates.

Sensors: We provide details about the recorded sensors in Tab. 4.2. The HoloLens has a specialized large field of view (FoV) multi-camera tracking rig (low resolution, global shutter) [334], while the iPad has a single, higher-resolution camera with rolling shutter and more limited FoV. All images are undistorted. We show samples of these images in Fig. 4.6. We also recorded outputs of the real-time AR tracking algorithms available on each device, which includes relative camera poses and sensor calibration. All sensor data is registered into a common reference frame with accurate absolute GT poses using the pipeline described in the next section.

4.4. Ground-truth generation

We estimate GT poses from the raw data from the different sensors. This process is fully automated and does not require any manual alignment or annotation.

Overview: We start by aligning different sessions of the laser scanner by using the images and the 3D LiDAR point cloud. When registered together, they form the GT reference map, which accurately captures the structure and appearance of the scene. We then register each AR sequence individually to the reference map using local feature matching and relative poses from the on-device tracker. Finally, all camera poses are refined jointly by optimizing the visual constraints within and across sequences.

Notation: We denote ${}_i\mathbf{T}_j \in \mathrm{SE}(3)$ the 6-DoF pose, encompassing rotation and

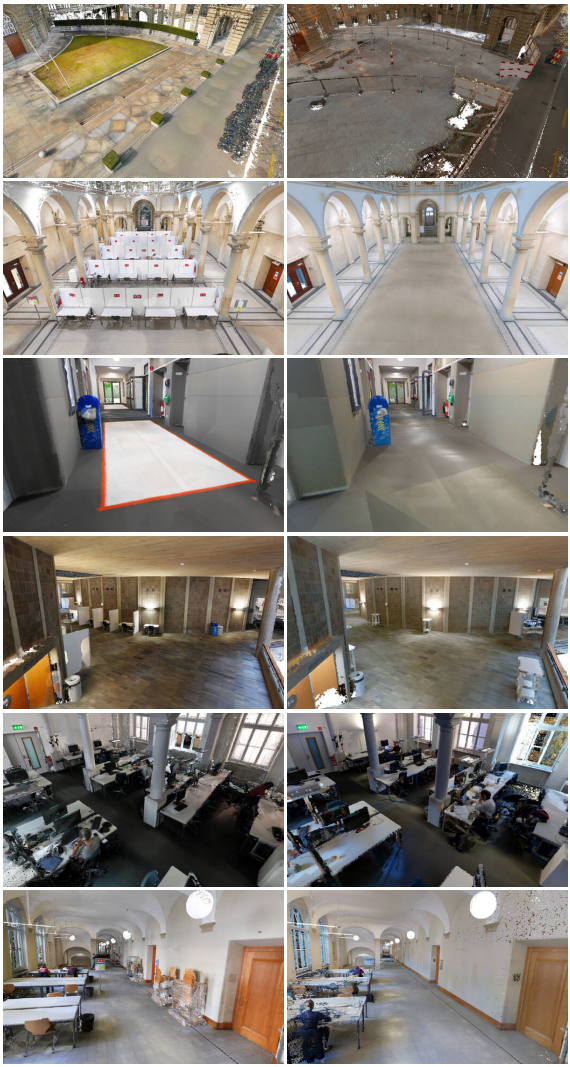


Figure 4.5.: ***Long-term structural changes.*** *LiDAR point clouds captured over a year reveal the geometric changes that spaces undergo at different time scales: 1) very rarely (construction work), 2-4) sparsely (displacement of furniture), or even 5-6) daily due to regular usage (people, objects).*

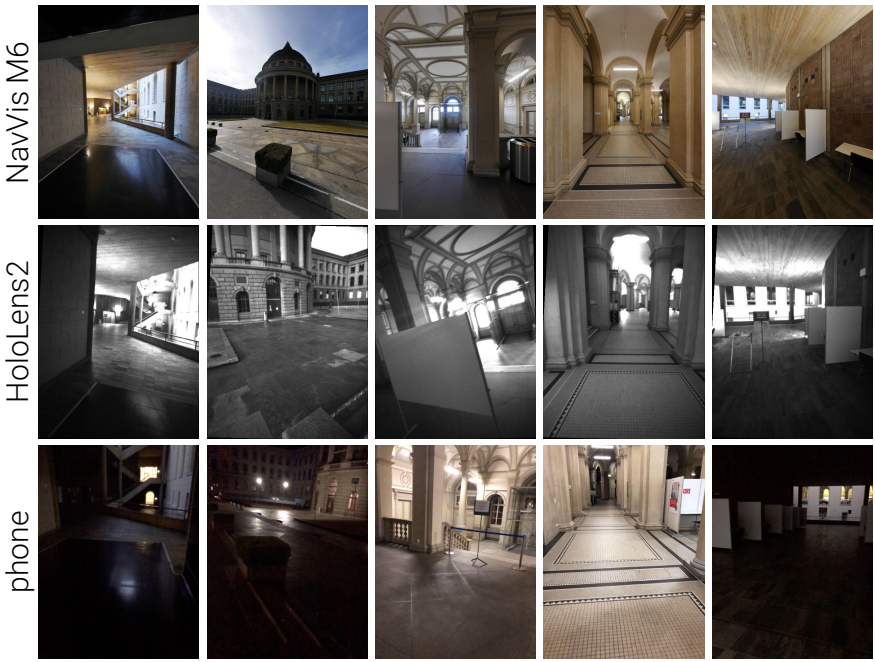


Figure 4.6.: ***Sample of images from the different devices:*** *NavVis M6, HoloLens2, phone. Each column shows a different scene of the HGE location with large illumination changes. NavVis and phone images are colored while HoloLens2 images are grayscale. NavVis images are always perfectly upright, while the viewpoint and height of HoloLens2 and phone images varies significantly. Despite the automatic exposure, phone images easily appear dark in night-time low-light conditions.*

translation, that transforms a point in frame j to another frame i. Our goal is to compute globally-consistent absolute poses ${}_w\mathbf{T}_i$ for all cameras i of all sequences and scanning sessions into a common reference world frame w.

4.4.1. Ground-truth reference model

Each capture session $S \in \mathcal{S}$ of the NavVis laser-scanning platform is processed by a proprietary inertial-LiDAR SLAM that estimates, for each image i, a pose ${}_0\mathbf{T}_i^S$ relative to the beginning of the session. The software filters out noisy LiDAR measurements, removes dynamic objects, and aggregates the remainder into a globally-consistent colored 3D point cloud with a grid resolution of 1cm. To recover visibility information, we compute a dense mesh using the Advancing Front algorithm [72].

Our first goal is to align the sessions into a common GT reference frame. We assume that the scan trajectories are drift-free and only need to register each with a rigid transformation ${}_w\mathbf{T}_0^S$. Scan sessions can be captured between extensive periods of time and therefore exhibit large structural and appearance changes. We use a combination of image and point cloud information to obtain accurate registrations without any manual initialization. The steps are inspired by the reconstruction pipeline of Choi et al. [63, 375].

Pair-wise registration: We first estimate a rigid transformation ${}_A\mathbf{T}_B$ for each pair of scanning sessions $(A, B) \in \mathcal{S}^2$. For each image I_i^A in A, we select the r most similar images $(I_j^B)_{1 \leq j \leq r}$ in B based on global image descriptors [10, 139, 244], which helps the registration scale to large scenes. We extract sparse local image features and establish 2D-2D correspondences $\{\mathbf{p}_i^A, \mathbf{p}_j^B\}$ for each image pair (i, j). The 2D keypoints $\mathbf{p}_i \in \mathbb{R}^2$ are lifted to 3D, $\mathbf{P}_i \in \mathbb{R}^3$, by tracing rays through the dense mesh of the corresponding session. This yields 3D-3D correspondences $\{\mathbf{P}_i^A, \mathbf{P}_j^B\}$, from which we estimate an initial relative pose [332] using RANSAC [95]. This pose is refined with the point-to-plane Iterative Closest Point (ICP) algorithm [255] applied to the pair of LiDAR point clouds.

We use state-of-the-art local image features that can match across drastic illumination and viewpoint changes [81, 245, 261]. Combined with the strong geometric constraints in the registration, our system is robust to long-term temporal changes

and does not require manual initialization. Using this approach, we have successfully registered building-scale scans captured at more than a year of interval with large structural changes.

Global alignment: We gather all pairwise constraints and jointly refine all absolute scan poses $\{{}_w\mathbf{T}_0^S\}$ by optimizing a pose graph [111]. The edges are weighted with the covariance matrices of the pair-wise ICP estimates. The images of all scan sessions are finally combined into a unique reference trajectory $\{{}_w\mathbf{T}_i^{\text{ref}}\}$. The point clouds and meshes are aligned according to the same transformations. They define the reference representation of the scene, which we use as a basis to obtain GT for the AR sequences.

Ground-truth visibility: The accurate and dense 3D geometry of the mesh allows us to compute accurate visual overlap between two cameras with known poses and calibration. Inspired by Rau et al. [241], we define the overlap of image i wrt. a reference image j by the ratio of pixels in i that are visible in j:

$$O(i \to j) = \frac{\sum_{k \in (W,H)} \mathbb{1}\left[\Pi_j({}_w\mathbf{T}_j, \Pi_i^{-1}({}_w\mathbf{T}_i, \mathbf{p}_k^i, z_k)) \in (W,H)\right] \alpha_k}{W \cdot H} \quad , \quad (4.1)$$

where Π_i projects a 3D point k to camera i, Π_i^{-1} conversely backprojects it using its known depth z_k with (W,H) as the image dimensions. The contribution of each pixel is weighted by the angle $\alpha_k = \cos(\mathbf{n}_{i,k}, \mathbf{n}_{j,k})$ between the two rays. To handle scale changes, it is averaged both ways $i \to j$ and $j \to i$. This score is efficiently computed by tracing rays through the mesh and checking for occlusion for robustness.

This score $O \in [0,1]$ favors images that observe the same scene from similar viewpoints. Unlike sparse co-visibility in an SfM model [238], our formulation is independent of the amount of texture and the density of the feature detections. This score correlates with matchability – we thus use it as GT when evaluating retrieval and to determine an upper bound on the theoretically achievable performance of our benchmark.

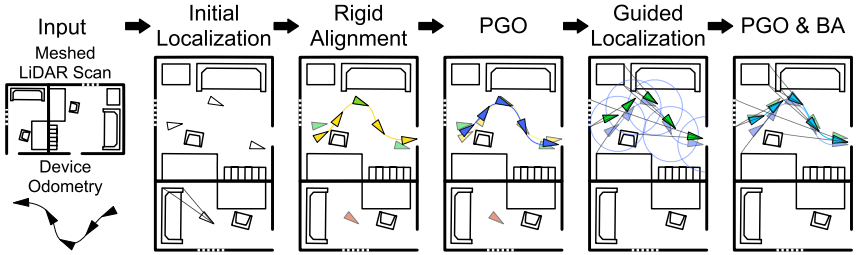


Figure 4.7.: ***Sequence-to-scan alignment.*** *We first estimate the absolute pose of each sequence frame using image retrieval and matching. This initial localization prior is used to obtain a single rigid alignment between the input trajectory and the reference 3D model via voting. The alignment is then relaxed by optimizing the individual frame poses in a pose graph based on both relative and absolute pose constraints. We bootstrap this initialization by mining relevant image pairs and re-localizing the queries. Given these improved absolute priors, we optimize the pose graph again and finally include reprojection errors of the visual correspondences, yielding a refined trajectory.*

4.4.2. Sequence-to-scan alignment

We now aim to register each AR sequence individually into the dense GT reference model (see Fig. 4.7). Given a sequence of n frames, we introduce a simple algorithm that estimates the per-frame absolute pose $\{{}_w\mathbf{T}_i\}_{1\leq i\leq n}$. A frame refers to an image taken at a given time or, when the device is composed of a camera rig with known calibration (e.g., HoloLens), to a collection of simultaneously captured images.

Inputs: We assume given trajectories $\{{}_0\mathbf{T}_i^{\text{track}}\}$ estimated by a visual-inertial tracker – we use ARKit for iPhone/iPad and the on-device tracker for HoloLens. The tracker also outputs per-frame camera intrinsics $\{\mathbf{C}_i\}$, which account for auto-focus or calibration changes and are for now kept fixed.

Initial localization: For each frame of a sequence $\{I_i^{\text{query}}\}$, we retrieve a fixed number r of relevant reference images $(I_j^{\text{ref}})_{1\leq j\leq r}$ using global image descriptors. We match sparse local features [81, 183, 245] extracted in the query frame to each retrieved image I_j^{ref} obtaining a set of 2D-2D correspondences $\{\mathbf{p}_{i,k}^{\text{q}}, \mathbf{p}_{j,k}^{\text{ref}}\}_k$. The 2D reference keypoints are lifted to 3D by tracing rays through the mesh of the reference model, yielding a set of 2D-3D correspondences $\mathcal{M}_{i,j} := \{\mathbf{p}_{i,k}^{\text{q}}, \mathbf{P}_{j,k}^{\text{ref}}\}_k$.

We combine all matches per query frame $\mathcal{M}_i = \cup_{j=1}^{r} \mathcal{M}_{i,j}$ and estimate an initial absolute pose ${}_w\mathbf{T}_i^{\text{loc}}$ using the (generalized) P3P algorithm [122] within a LO-RANSAC scheme [68] followed by a non-linear refinement [278]. Because of challenging appearance conditions, structural changes, or lack of texture, some frames cannot be localized in this stage. We discard all poses that are supported by a low number of inlier correspondences.

Rigid alignment: We next recover a coarse initial pose $\{{}_w\mathbf{T}_i^{\text{init}}\}$ for all frames, including those that could not be localized. Using the tracking, which is for now assumed drift-free, we find the rigid alignment ${}_w\mathbf{T}_0^{\text{init}}$ that maximizes the consensus among localization poses. This voting scheme is fast and effectively rejects poses that are incorrect, yet confident, due to visual aliasing and symmetries. Each estimate is a candidate transformation ${}_w\mathbf{T}_0^i = {}_w\mathbf{T}_i^{\text{loc}} \left({}_0\mathbf{T}_i^{\text{track}}\right)^{-1}$, for which other frames can vote, if they are consistent within a threshold τ_{rigid}. We select the candidate with the highest count of inliers:

$$ {}_w\mathbf{T}_0^{\text{init}} = \underset{\mathbf{T} \in \{{}_w\mathbf{T}_0^i\}_{1 \le i \le n}}{\arg\max} \sum_{1 \le j \le n} \mathbb{1}\left[\text{dist}\left({}_w\mathbf{T}_j^{\text{loc}}, \mathbf{T} \cdot {}_0\mathbf{T}_j^{\text{track}}\right) < \tau_{\text{rigid}}\right] \quad , \tag{4.2} $$

where $\mathbb{1}[\cdot]$ is the indicator function and $\text{dist}(\cdot,\cdot)$ returns the magnitude, in terms of translation and rotation, of the difference between two absolute poses. We then recover the per-frame initial poses as $\{{}_w\mathbf{T}_i^{\text{init}} := {}_w\mathbf{T}_0^{\text{init}} \cdot {}_0\mathbf{T}_i^{\text{track}}\}_{1 \le i \le n}$.

Pose graph optimization: We refine the initial absolute poses by maximizing the consistency of tracking and localization cues within a pose graph. The refined poses $\{{}_w\mathbf{T}_i^{\text{PGO}}\}$ minimize the energy function

$$ E(\{{}_w\mathbf{T}_i\}) = \sum_{i=1}^{n-1} \mathcal{C}_{\text{PGO}}\left({}_w\mathbf{T}_{i+1}^{-1}\, {}_w\mathbf{T}_i,\ {}_{i+1}\mathbf{T}_i^{\text{track}}\right) + \sum_{i=1}^{n} \mathcal{C}_{\text{PGO}}\left({}_w\mathbf{T}_i,\ {}_w\mathbf{T}_i^{\text{loc}}\right) \quad , \tag{4.3} $$

where $\mathcal{C}_{\text{PGO}}(\mathbf{T}_1, \mathbf{T}_2) := \left\|\text{Log}\left(\mathbf{T}_1\, \mathbf{T}_2^{-1}\right)\right\|_{\Sigma,\gamma}^2$ is the distance between two absolute or relative poses, weighted by covariance matrix $\Sigma \in \mathbb{R}^{6\times6}$ and loss function γ. Here, Log maps from the Lie group $\text{SE}(3)$ to the corresponding algebra $\mathfrak{se}(3)$.

We robustify the absolute term with the Geman-McClure loss function and anneal its scale via a Graduated Non-Convexity scheme [359]. This ensures convergence

in case of poor initialization, e.g., when the tracking exhibits significant drift, while remaining robust to incorrect localization estimates. The covariance of the absolute term is propagated from the preceding non-linear refinement performed during localization. The covariance of the relative term is recovered from the odometry pipeline, or, if not available, approximated as a factor of the motion magnitude.

This step can fill the gaps from the localization stage using the tracking information and conversely correct for tracker drift using localization cues. In rare cases, the resulting poses might still be inaccurate when both the tracking drifts and the localization fails.

Guided localization via visual overlap: To further increase the pose accuracy, we leverage the current pose estimates $\{{}_w\mathbf{T}_i^{\text{PGO}}\}$ to mine for additional localization cues. Instead of relying on global visual descriptors, which are easily affected by aliasing, we select reference images with a high overlap using the score defined in Sec. 4.4.1. For each sequence frame i, we select r reference images with the largest overlap and again match local features and estimate an absolute pose. These new localization priors improve the pose estimates in a second optimization of the pose graph.

Bundle adjustment: For each frame i, we recover the set of 2D-3D correspondences $\mathcal{M}_i$ used by the guided re-localization. We now refine the poses $\{{}_w\mathbf{T}_i^{\text{BA}}\}$ by jointly minimizing a bundle adjustment problem with relative pose graph costs:

$$
\begin{aligned}
E(\{{}_w\mathbf{T}_i\}) = &\sum_{i=1}^{n-1} \mathcal{C}_{\text{PGO}} \left({}_w\mathbf{T}_{i+1}^{-1} \; {}_w\mathbf{T}_i, \; {}_{i+1}\mathbf{T}_i^{\text{track}} \right) \\
&+ \sum_{i=1}^{n} \sum_{\mathcal{M}_{i,j} \in \mathcal{M}_i} \sum_{(\mathbf{p}_k^{\text{ref}}, \mathbf{P}_k^{\text{q}}) \in \mathcal{M}_{i,j}} \left\| \Pi({}_w\mathbf{T}_i, \mathbf{P}_{j,k}^{\text{ref}}) - \mathbf{p}_{i,k}^{\text{q}} \right\|_{\sigma^2}^2 ,
\end{aligned}
\tag{4.4}
$$

where the second term evaluates the reprojection error of a 3D point $\mathbf{P}_{j,k}^{\text{ref}}$ for observation k to frame i. The covariance is the noise σ^2 of the keypoint detection algorithm. We pre-filter correspondences that are behind the camera or have an initial reprojection error greater than $\sigma\,\tau_{\text{reproj}}$. As the 3D points are sampled from the LiDAR, we also optimize them with a prior noise corresponding to the LiDAR specifications. We use the Ceres [3] solver.

4.4.3. Joint global refinement

Once all sequences are individually aligned, we refine them jointly by leveraging sequence-to-sequence visual observations. This is helpful when sequences observe parts of the scene not mapped by the LiDAR. We first triangulate a sparse 3D model from scan images, aided by the mesh. We then triangulate additional observations, and finally jointly optimize the whole problem.

Reference triangulation: We estimate image correspondences of the reference scan using pairs selected according to the visual overlap defined in Sec. 4.4.2. Since the image poses are deemed accurate and fixed, we filter the correspondences using the known epipolar geometry. We first consider feature tracks consistent with the reference surface mesh before triangulating more noisy observations within LO-RANSAC using COLMAP [278]. The remaining feature detections, which could not be reliably matched or triangulated, are lifted to 3D by tracing through the mesh. This results in an accurate, sparse SfM model with tracks across reference images.

Sequence optimization: We then add each sequence to the sparse model. We first establish correspondences between images of the same and of different sequences. The image pairs are again selected by highest visual overlap computed using the aligned poses $\{{}_w\mathbf{T}_i^{\mathrm{BA}}\}$. The resulting tracks are sequentially triangulated, merged, and added to the sparse model. Finally, all 3D points and poses are jointly optimized by minimizing the joint pose-graph and bundle adjustment (Eq. (4.4)). As in COLMAP [278], we alternate optimization and track merging. To scale to large scenes, we subsample keyframes from the full frame-rate captures and only introduce absolute pose and reprojection constraints for keyframes while maintaining all relative pose constraints from tracking.

4.4.4. Ground-truth validation

Potential limits: Brachmann et al. [34] observe that algorithms generating pseudo-GT poses by minimizing either 2D or 3D cost functions alone can yield noticeably different results. We argue that there exists a single underlying, true GT. Reaching it requires fusing large amounts of redundant data with sufficient sensors of sufficiently low noise. Our GT poses optimize complementary constraints from visual and

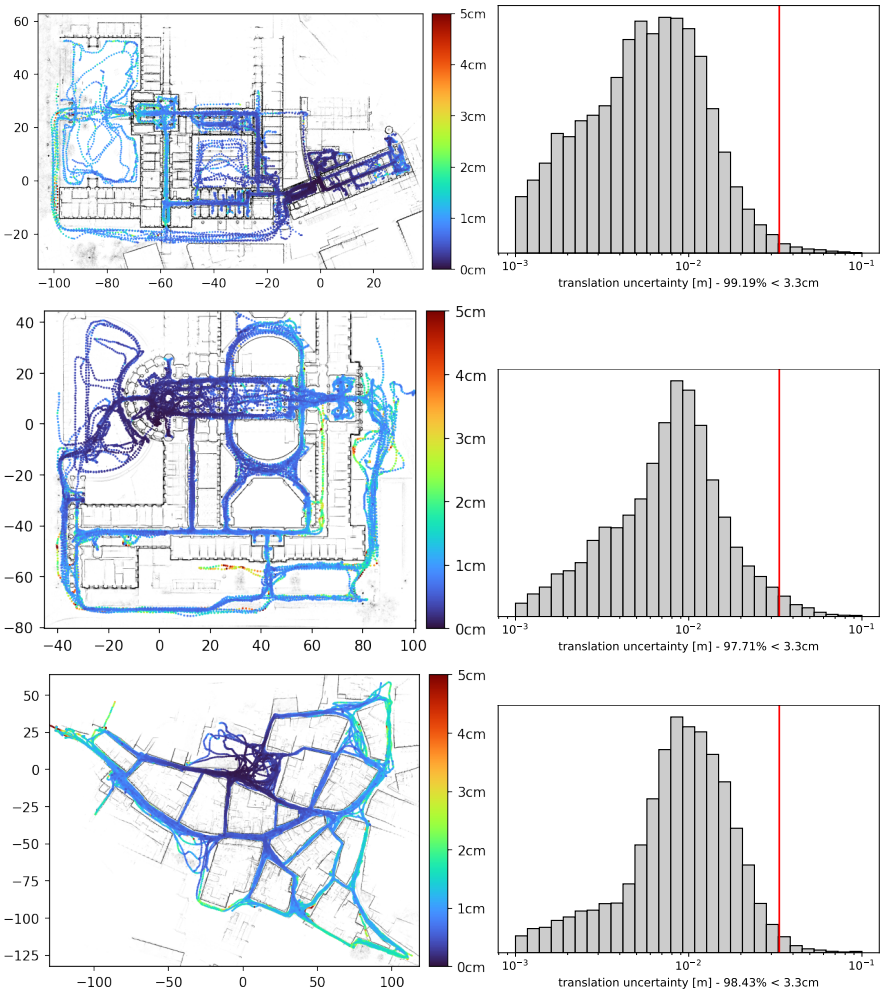


Figure 4.8.: ***Translation uncertainties of the ground truth camera centers*** *for the CAB (top), LIN (middle) and HGE (bottom) scenes. Left: The overhead map shows that the uncertainties are larger in areas that are not well covered by the 3D scanners or where the scene is further away from the camera, such as in long corridors and large outdoor space. Right: The histogram of uncertainties shows that most images have an uncertainty far lower than* σ_t=3.33*cm.*

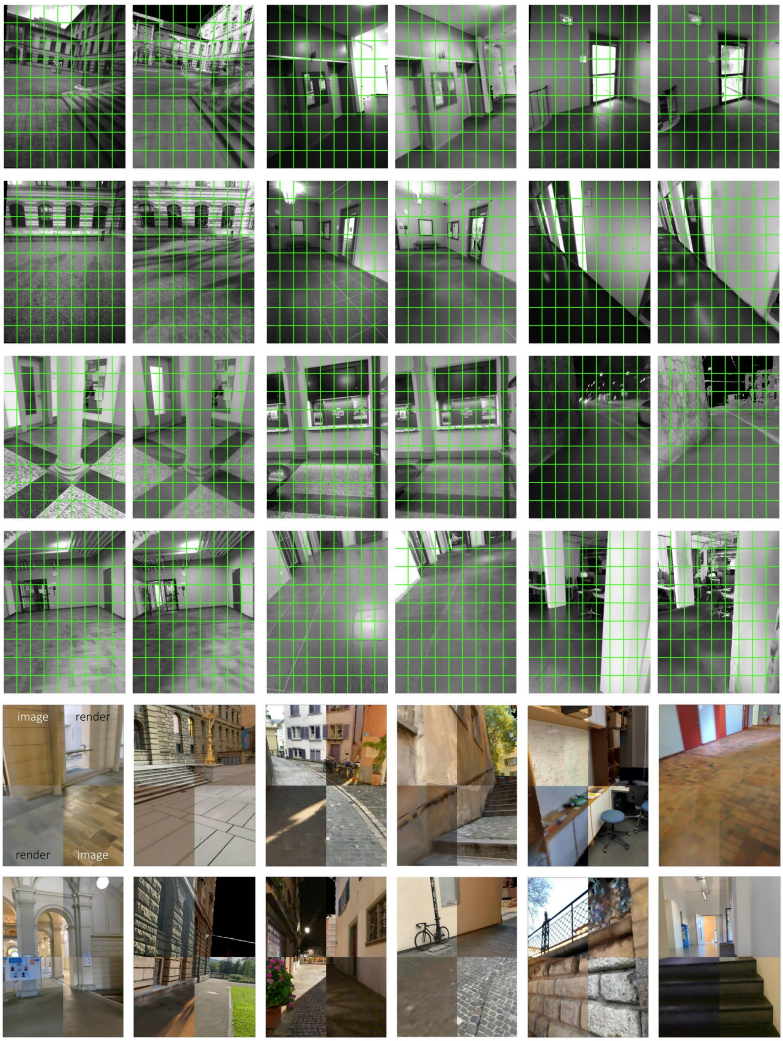


Figure 4.9.: ***Qualitative renderings from the mesh.*** *Top: We render images at the ground-truth poses from the vertex-colored mesh (right) and compare them to the originals (left). We show 6 HoloLens images in the first two rows and six phone images in the next two. We overlay a regular grid to facilitate the comparison. Bottom: We show mosaics that combine the originals (top-left, bottom-right) and the renderings (top-right, bottom-left).*

inertial measurements, guided by an accurate LiDAR-based 3D structure. Careful design and propagation of uncertainties reduces the bias towards one of the sensors. All sensors are factory- and self-calibrated during each recording by the respective commercial, production-grade SLAM algorithms. We do not claim that our GT is perfect but analyzing the optimization uncertainties sheds light on its degree of accuracy.

Pose uncertainty: We estimate the uncertainties of the GT poses by inverting the Hessian of the refinement. To obtain calibrated covariances, we scale them by the empirical keypoint detection noise, estimated as σ=1.33 pixels for the CAB scene. The maximum noise in translation is the size of the major axis of the uncertainty ellipsoids, which is the largest eivenvalue σ_t^2 of the covariance matrices. Fig. 4.8 shows its distribution for the CAB scene. We retain images whose poses are correct within 10cm with a confidence of 99.7%. For normally distributed errors, this corresponds to a maximum uncertainty σ_t=3.33cm and discards 0.8% of all frames. For visual inspection, we render images at the estimated GT camera poses using the colored mesh. As shown in Fig. 4.9, they appear pixel-aligned with the original images, supporting that the poses are accurate.

4.4.5. Selection of mapping and query sequences

We divide the set of sequences into two disjoint groups for mapping and localization. Mapping sequences are selected such that they have a minimal overlap between each other yet cover the area visited by all remaining sequences. This simulates a scenario of minimal coverage and maximizes the number of localization query sequences.

Algorithm: Let $C(i,j)_k$ be the coverage, a boolean that indicates whether the image k of sequence i shares sufficient covisibility with at least one image is sequence j. Here two images are deemed covisible if they co-observe a sufficient number of 3D points in the final, full SfM sparse model [238] or according to the ground truth mesh-based visual overlap. The coverage of sequence i with a set of other sequences $\mathcal{S} = \{j\}$ is the ratio of images in i that are covered by at least one

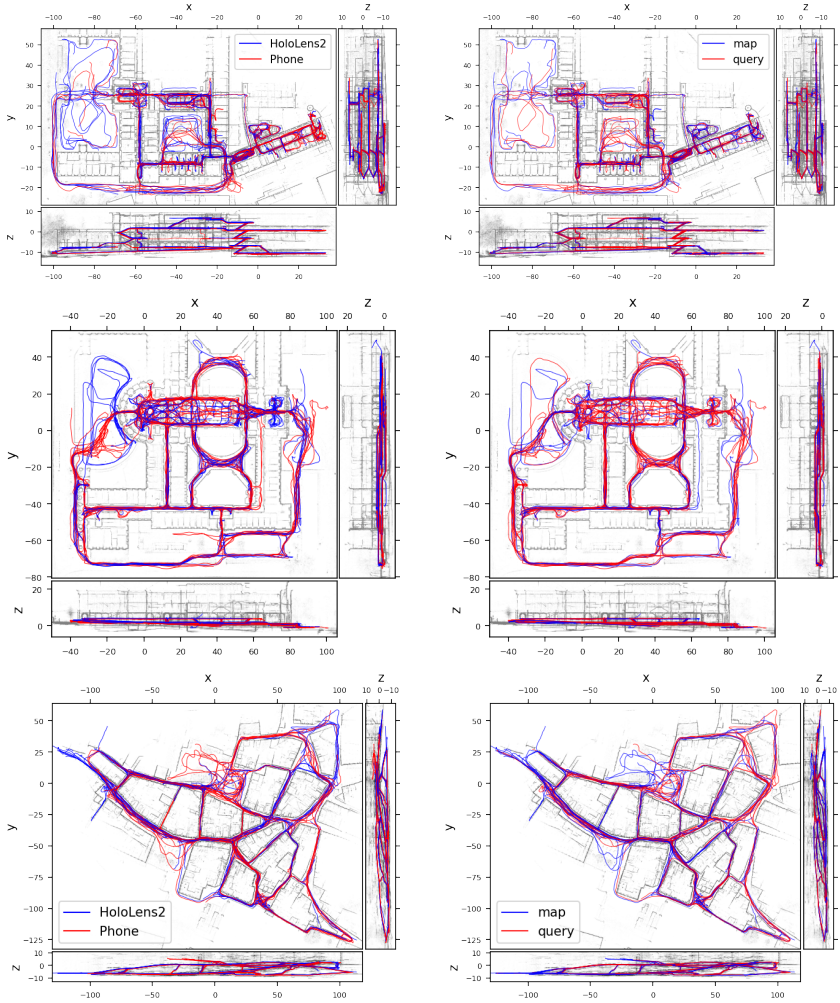


Figure 4.10.: ***Spatial distribution of sequences*** *for the CAB (top), HGE (middle), and LIN (bottom) locations. We show the ground truth trajectories overlaid on the LiDAR point clouds along 3 orthogonal directions. All axes are in meters and z is aligned with the gravity. Left: Types of devices among all registered sequences. Right: Map and query sequences selected for evaluation. CAB spans multiple floors while HGE and LIN are mostly 2D but include a range of ground heights. The space is well covered by both types of devices and sequences.*

image in $\mathcal{S}$:

$$C(i, \mathcal{S}) = \frac{1}{|i|} \sum_{k \in i} \bigcap_{j \in \mathcal{S}} C(i, j)_k \tag{4.5}$$

We seek to find the set of mapping sequences $\mathcal{M}$ and remaining query sequences $\mathcal{Q} = \mathcal{S} \backslash \mathcal{M}$ that minimize the coverage between map sequences while ensuring that each query is sufficiently covered by the map:

$$\begin{aligned} \mathcal{M}^* = \arg\min \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} C(i, \mathcal{M} \backslash \{i\}) \\ \text{such that} \quad C(i, \mathcal{M}) > \tau \quad \forall i \in \mathcal{Q} \ , \end{aligned} \tag{4.6}$$

where τ is the minimum query coverage. We ensure that query sequences are out of coverage for at most t consecutive seconds, where t can be tuned to adjust the difficulty of the localization and generally varies from 1 to 5 seconds. This problem is combinatorial and without exact solution. We solve it approximately with a best-first search that iteratively adds new images and checks for the feasibility of the solution. At each step, we consider the query sequences that are the least covisible with the current map.

Data distribution: We enforce that night-time sequences are not included in the map, which is a realistic assumption for crowd-sourced scenarios. We do not enforce an equal distribution of device types in either group but observe that this occurs naturally. For the evaluation, mapping images are sampled at intervals of at most 2.5FPS, 50cm of distance, and 20°of rotation. This ensures a sufficient covisibility between subsequent frames while reducing the computational cost of creating maps. The queries are sampled every 1s/1m/20°and, for each device type, 1000 poses are randomly selected out of those with sufficiently low uncertainty. The resulting distributions are shown in Fig. 4.10.

4.5. Evaluation

We evaluate state-of-the-art approaches in both single-frame and sequence settings and summarize our results in Fig. 4.11. All results are averaged across all locations.

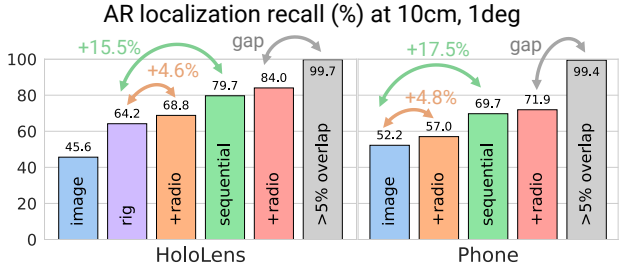


Figure 4.11.: ***Main results.*** *We show results for Fusion image retrieval with SuperPoint local features and SuperGlue matcher on both HoloLens 2 and phone queries. We consider several tracks: single-image / single-rig localization with / without radios and similarly for sequence (10 seconds) localization. In addition, we report the percentage of queries with at least 5% ground-truth overlap with respect to the best mapping image.*

Single-frame: We first consider in Sec. 4.5.1 the classical academic setup of single-frame queries (single image for phones and single rig for HoloLens 2) without additional sensor. We then look at how radio signals can be beneficial. We also analyze the impact of various settings: FoV, type of mapping images, and mapping algorithm.

Sequence: Second, by leveraging the real-time AR tracking poses, we consider the problem of sequence localization in Sec. 4.5.2. This corresponds to a real-world AR application retrieving the content attached to a target map using the real-time sensor stream from the device. In this context, we care not only about accuracy and recall but also about the time required to localize accurately, which we call the *time-to-recall*.

4.5.1. Single-frame localization

We first evaluate several algorithms representative of the state of the art in the classical single-frame academic setup. We consider the hierarchical localization framework with different approaches for image retrieval and matching. Each of them first builds a sparse SfM map from reference images. For each query frame, we then retrieve relevant reference images, match their local features, lift the reference keypoints to 3D using the sparse map, and finally estimate a pose with PnP+RANSAC.

Hierarchical localization		Query device	
Retrieval	Matching	HL2	Phone
NetVLAD	SIFT+AdaLAM	48.3 / 63.7	38.0 / 54.8
	DoG+SOSNet	52.3 / 67.3	37.9 / 55.4
	R2D2	48.2 / 63.9	42.1 / 58.4
	SP+SG	59.9 / 73.0	50.1 / 63.3
Fusion	SIFT+AdaLAM	51.2 / 67.9	38.5 / 56.9
	DoG+SOSNet	55.2 / 71.2	39.3 / 57.4
	R2D2	52.0 / 68.4	43.5 / 60.2
	SP+SG	64.2 / 77.4	52.2 / 65.8

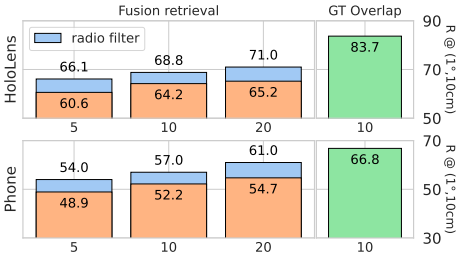


Table 4.3.: ***Left: single-frame localization.*** *We report the recall at* $(1^\circ, 10cm)/(5^\circ, 1m)$ *for baselines representative of the state of the art. Our dataset is challenging while most others are saturated. There is a clear progress from SIFT but also large room for improvement.* ***Right: localization with radio signals.*** *Increasing the number* $\{5, 10, 20\}$ *of retrieved images increases the localization recall at* $(1^\circ, 10cm)$*. The best-performing visual retrieval (Fusion, orange) is however far worse than the GT overlap. Filtering with radio signals (blue) improves the performance in all settings.*

We report the recall of the final pose at two thresholds [273]: 1) a fine threshold at $(1^\circ, 10\text{cm})$, which we see as the minimum accuracy required for a good AR user experience in most settings. 2) a coarse threshold at $(5^\circ, 1\text{m})$ to show the room for improvement for current approaches.

We evaluate global descriptors computed by NetVLAD [10] and by a fusion [135] of NetVLAD and APGeM [244], which are representative of the field [233]. We retrieve the 10 most similar images. For matching, we evaluate handcrafted SIFT [183], SOSNet [317] as a learned patch descriptor extracted from DoG [183] keypoints, and a robust deep-learning based joint detector and descriptor R2D2 [245]. SIFT features are matched by AdaLAM [54] and the others by exact mutual nearest neighbor search. We also evaluate SuperGlue [263] – a learned matcher based on SuperPoint [81] features. To build the map, we retrieve neighboring images filtered by frustum intersection from reference poses, match these pairs, and triangulate a sparse SfM model using COLMAP [278].

We report the results in Tab. 4.3 (left). Even the best methods have a large gap to perfect scores and much room for improvement. In the remaining ablation, we solely rely on SuperPoint+SuperGlue [81, 263] for matching as it clearly performs the best.

Leveraging radio signals: In this experiment, we show that radio signals can be used to constrain the search space for image retrieval. This has two main benefits: 1) it reduces the risk of incorrectly considering visual aliases, and 2) it lowers the compute requirements by reducing that numbers of images that need to be retrieved and matched. We implement this filtering as follows. We first split the scene into a sparse 3D grid considering only voxels containing at least one mapping frame. For each frame, we gather all radio signals in a $\pm$2s window and associate them to the corresponding voxel. If the same endpoint is observed multiple times in a given voxel, we average the received signal strengths (RSSI) in dBm. For a query frame, we similarly aggregate signals over the past 2s and rank voxels by their L2 distance between RSSIs, considering those with at least one common endpoint. We thus restrict image retrieval to 2.5% of the map.

Table 4.3 (right) shows that radio filtering always improves the localization accuracy over vanilla vision-only retrieval, irrespective of how many images are matches. The upper bound based on the GT overlap, defined in Sec. 4.4.1, shows that there is still much room for improvement for both image and radio retrieval. As the GT overlap baseline is far from the perfect 100% recall, frame-to-frame matching and pose estimation have also much room to improve.

Varying field-of-view: We study the impact of the FoV of the HoloLens 2 device via two configurations: 1) Each camera in a rig is seen as a single-frame and localized using LO-RANSAC + P3P. 2) We consider all four cameras in a frame and localize them together using the generalized solver GP3P. With fusion retrieval, SuperPoint, and SuperGlue, single images (1) only achieve 45.6% / 61.3% recall, while using rigs (2) yields 64.2% / 77.4% (Tab. 4.3). Rig localization is thus highly beneficial, especially in hard cases where single cameras face texture-less areas, such as the ground and walls.

Mapping modality: We study whether the high-quality LiDAR mesh can be used for localization. We consider two approaches to obtain a sparse 3D point cloud: 1) By triangulating sparse visual correspondences across multiple views. 2) By lifting 2D keypoints in reference images to 3D by tracing rays through the mesh. Lifting can leverage dense correspondences, which cannot be efficiently triangulated with conventional multi-view geometry. We thus compare 1) and 2) with SuperGlue to 2) with LoFTR [303], a state-of-the-art dense matcher. The results in Tab. 4.4 (right)

Mapping images →		HL2 + Phone		HD 360	Both
Image pairs from →		Retrieval	GT	Retrieval	Retrieval
Matching	Device	+ Poses	overlap	+ Poses	+ Poses
SP + SG	HL2	64.2 / 77.4	64.2 / 77.3	70.1 / 83.6	64.1 / 77.5
	Phone	52.2 / 65.8	52.9 / 66.3	47.4 / 64.9	60.6 / 72.1

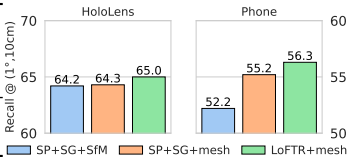


Table 4.4.: ***Impact of mapping. Left: Scenarios.*** *Building the map with HD 360 images from NavVis scanners, instead of or with dense AR sequences, does not consistently boost the performance as they are usually sparser, do not fully cover each location, and have different characteristics than AR images.* ***Right: Modalities.*** *Lifting 2D points to 3D using the LiDAR mesh instead of triangulating with SfM is beneficial. This can also leverage dense matching, e.g., with LoFTR.*

Condition	CAB scene		HGE scene		LIN scene
	Indoor	Outdoor	Indoor	Outdoor	Outdoor
day	66.5 / 74.7	73.9 / 88.1	52.7 / 65.9	43.0 / 64.3	71.2 / 82.5
night	30.3 / 44.8	18.8 / 40.6	47.9 / 59.4	12.1 / 33.6	38.6 / 55.6

Table 4.5.: ***Impact of the condition and environment*** *on single-image phone localization. During the day, localizing indoors can be more accurate (10cm threshold) but less robust (1m threshold) than outdoors due to visual aliasing and a lack of texture. Night-time localization is more challenging outdoors than indoors because of a larger drop of illumination.*

show that the mesh brings some improvements. Points could also be lifted by dense depth from multi-view stereo. We however did not obtain satisfactory results with a state-of-the-art approach [345] as it cannot handle very sparse mapping images.

Mapping scenario: We study the accuracy of localization against maps built from different types of images: 1) crowd-sourced, dense AR sequences; 2) curated, sparser HD 360 images from the NavVis device; 3) a combination of the two. The results are summarized in Tab. 4.4 (left), showing that the mapping scenario has a large impact on the final numbers. On the other hand, image pair selection for mapping matters little. Crowd-sourcing and manual scans can complement each other well to address an imperfect scene coverage. We hope that future work can close the gap between the scenarios to achieve better metrics from crowd-sourced data without curation.

Condition and environment: We now investigate the impact of different capture conditions (day, night) and environment (indoor, outdoor) of the query images. Query sequences are labeled as day or night based on the time and date of capture. We manually annotate overhead maps into indoor and outdoor areas. We report the results for single-image localization of phone images in Tab. 4.5.

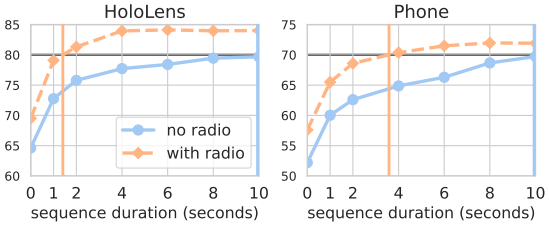
In regular day-time conditions, outdoor areas exhibit distinctive texture and are thus easier to coarsely localize in than texture-less, repetitive indoor areas. The scene structure is however generally further away from the camera, so optimizing reprojection errors yields less accurate camera poses.

Indoor scenes generally benefit from artificial light and are thus minimally affected by the night-time drop of natural light. Outdoor scenes benefit from little artificial light, mostly due to sparse street lighting, and thus widely change in appearance between day and night. As a result, the localization performance drops to a larger extent outdoors than indoors.

4.5.2. Sequence localization

In this section, inspired by typical AR use cases, we consider the problem of sequence localization, which aims to align multiple consecutive frames using sensor data aggregated over short time intervals.

Approach: Our baseline for this task is based on the ground-truthing pipeline and has as such relatively high compute requirements. However, we are primarily interested in demonstrating the potential performance gains by leveraging multiple frames. First, we run image retrieval and single-frame localization, followed by a first pose graph optimization (PGO) with tracking and localization poses. Then, we do a second localization with retrieval guided by the poses of the first PGO, followed by a second PGO. Finally, we run a pose refinement by considering reprojections to query frames and tracking cost. We can also use radio signals to restrict image retrieval throughout the pipeline. As previously, we consider the localization recall but only of the last frame in each sequence, which is the one that influences the current AR user experience in a real-time scenario.



Device	Radios?	TTR@X%		
		70%	80%	90%
HL2	✗	<1s	>10s	>10s
	✓	<1s	1.40s	>10s
Phone	✗	>10s	>10s	>10s
	✓	3.58s	>10s	>10s

Figure 4.12.: ***Sequence localization.*** *We report the localization recall at* $(1^\circ, 10cm)$ *of SuperPoint features with SuperGlue matcher as we increase the duration of each sequence. The pipeline leverages both on-device tracking and absolute localization, as vision-only (solid) or combined with radio signals (dashed). We show the time-to-recall (TTR) at 80% for HL2 and at 70% for phone queries. Using radio signals reduces the TTR from over 10s to 1.40s and 3.58s, respectively.*

Results: We evaluate various query durations and introduce the *time-to-recall* metric as the sequence length (time) required to successfully localize X% (recall) of the queries within (1°, 10cm), or, in short, TTR@X%. Localization algorithms should aim to minimize this metric to render retrieved content as quickly as possible after starting an AR experience. Fig. 4.12 reports the results averaged over all locations. While the performance of current methods is not satisfactory yet to achieve a TTR@90% under 10 seconds, using sequence localization leads to significant gains of 20%. The radio signals improve the performance in particular with shorter sequences and thus effectively reduce the time-to-recall.

Ablation: We ablate the different parts of our proposed sequence localization pipeline on sequences of 20 seconds. We report in Tab. 4.6 the localization recall at $\{1^\circ, 10\text{cm}\}$ and $\{5^\circ, 1\text{m}\}$ for both HoloLens 2 and Phone queries. The initial PGO with tracking and absolute constraints already offers a significant boost in performance compared to single-frame localization. We notice that the re-localization with image retrieval guided by the PGO poses achieves much better results than the first localization – this points to retrieval being a bottle-neck, not feature matching. Next, the second PGO is able to leverage the improved absolute constraints and yields better results. Finally, the pose refinement optimizing reprojection errors while also taking into account tracking constraints further improves the performance, notably at the tighter threshold.

Device	Radios	Steps					
		Loc.	Init.	PGO1	Re-loc.	PGO2	BA
HL2	✗	66.0 / 79.9	66.1 / 92.5	71.8 / 92.4	74.2 / 88.0	74.9 / 92.5	**79.3 / 92.8**
	✓	67.7 / 82.3	66.4 / 94.5	74.3 / 94.3	76.2 / 90.1	76.7 / 94.4	**81.6 / 94.9**
Phone	✗	54.2 / 65.5	52.4 / 88.0	62.7 / 87.7	61.8 / 77.4	66.1 / 88.4	**69.0 / 88.6**
	✓	56.7 / 71.5	54.1 / **90.2**	64.4 / 89.8	63.1 / 79.5	66.9 / 90.1	**71.0 / 90.2**

Table 4.6.: ***Ablation of the sequence localization.*** *We report recall for the different steps of the sequence localization pipeline for 10s sequences on the CAB location. The second localization, guided by the poses of the first PGO, drastically improves over the initial localization, especially when no radio signals are used. The final pose refinement optimizing reprojection errors while also taking into account tracking constraints offers a significant boost for the tighter threshold.*

4.6. Summary and outlook

Summary: LaMAR is the first benchmark that faithfully captures the challenges and opportunities of AR for visual localization and mapping. We first identified several key limitations of current benchmarks that make them unrealistic for AR. To address these limitations, we developed a new ground-truthing pipeline to accurately and robustly register AR sensor streams in large and diverse scenes aided by laser scans without any manual labeling or custom infrastructure. With this new benchmark, initially covering 3 large locations, we revisited the traditional academic setup and showed a large performance gap for existing state-of-the-art methods when evaluated using more realistic and challenging data.

We implemented simple yet representative baselines to take advantage of the AR-specific setup and we presented new insights that pave promising avenues for future works. We showed the large potential of leveraging other sensor modalities like radio signals, depth, or query sequences instead of single images. We also hope to direct the attention of the community towards improving map representations for crowd-sourced data and towards considering the time-to-recall metric, which is currently largely ignored. We publicly release at `lamar.ethz.ch` the complete LaMAR dataset, our ground-truthing pipeline, and the implementation of all baselines. The evaluation server and public leaderboard facilitates the benchmarking of

new approaches to keep track of the state of the art. We hope this will spark future research addressing the challenges of AR.

Limitations and impact: The large scale of LaMAR and its extensive coverage of multiple challenges is also its weakness: the evaluation requires significantly more compute resources than previous benchmarks and it is much harder to draw insights from the data given its sheer size. Dividing the dataset into smaller evaluation units could possibly make it easier to evaluate other tasks like image retrieval or SfM.

Because it includes only three scenes with relatively limited visual diversity, it is unclear whether LaMAR dataset can be used for the training of learning algorithms, for which the training data is currently the main bottleneck. As such, LaMAR has had a relatively small impact of the development of new algorithms for multi-sensor mapping and localization. We believe that the data could however be useful as auxiliary training data for other tasks like monocular depth or normal estimation.

Finally, while the GT poses are more accurate than found in existing localization datasets, they are not sufficiently accurate to evaluate high-accuracy algorithms like visual-inertial SLAM. In order to increase the accuracy of the poses, we believe that the GT pipeline should include inertial measurements and ground control points measured by survey devices.

Part II.

Leveraging 2D Maps

CHAPTER 5

Visual Localization in 2D Public Maps with Neural Matching

Humans can orient themselves in their 3D environments using simple 2D maps. Differently, as seen in Part I, algorithms for visual localization mostly rely on complex 3D point clouds that are expensive to build, store, and maintain over time. We bridge this gap by introducing OrienterNet, a deep neural network that can localize an image with sub-meter accuracy using the same 2D semantic maps that humans use. OrienterNet estimates the location and orientation of a query image by matching a neural Bird's-Eye View with open and globally available maps from OpenStreetMap, enabling anyone to localize anywhere such maps are available. OrienterNet is supervised only by camera poses but learns to perform semantic matching with a wide range of map elements in an end-to-end manner. To enable this, we introduce a large crowd-sourced dataset of images captured across 12 cities from the diverse viewpoints of cars, bikes, and pedestrians. OrienterNet generalizes to new datasets and pushes the state of the art in both robotics and AR scenarios.

5.1. Introduction

As humans, we intuitively understand the relationship between what we see and what is shown on a map of the scene we are in. When lost in an unknown area,

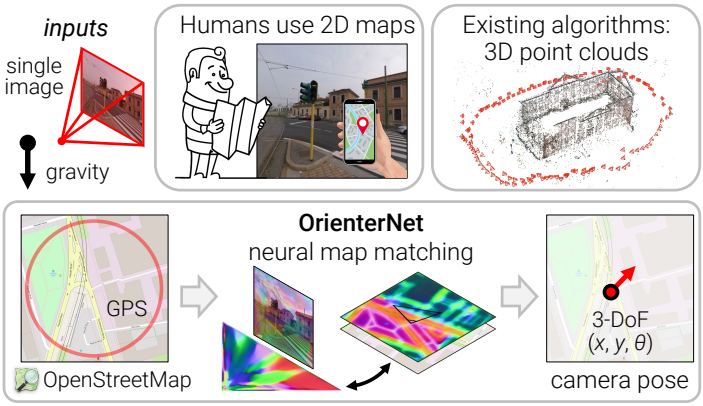


Figure 5.1.: ***Towards human-like localization.*** *Humans can easily orient themselves with basic 2D maps while state-of-the-art algorithms for visual localization require complex 3D cues. OrienterNet can localize an image using only compact maps from OpenStreetMap by matching Bird's-Eye View and neural maps.*

we can accurately pinpoint our location by carefully comparing the map with our surroundings using distinct geographic features.

Yet, algorithms for accurate visual localization are typically complex, as they rely on image matching and require detailed 3D point clouds and visual descriptors [81, 138, 183, 187, 253, 261, 271]. Building 3D maps with LiDAR or photogrammetry [2, 99, 205, 278, 294] is expensive at world scale and requires costly, freshly-updated data to capture temporal changes in visual appearance. 3D maps are also expensive to store, as they are orders of magnitude larger than basic 2D maps. This prevents executing localization on-device and usually requires costly cloud infrastructure. Spatial localization is thus a serious bottleneck for the large-scale deployment of robotics and augmented reality devices. This disconnect between the localization paradigms of humans and machines leads to the important research question of *How can we teach machines to localize from basic 2D maps like humans do?*

This chapter introduces an approach that can localize single images and image sequences with sub-meter accuracy given the same maps that humans use. These *planimetric* maps encode only the location and coarse 2D shape of few important objects but not their appearance nor height. Such maps are extremely compact,

up to 10^4 times smaller in size than 3D maps, and can thus be stored on mobile devices and used for on-device localization within large areas. We demonstrate these capabilities with OpenStreetMap (OSM) [218], an openly accessible and community-maintained world map, enabling anyone to localize anywhere for free. This solution does not require building and maintaining costly 3D maps over time nor collecting potentially sensitive mapping data.

Concretely, our algorithm estimates the 3-DoF pose, as position and heading, of a calibrated image in a 2D map. The estimate is probabilistic and can therefore be fused with an inaccurate GNSS prior or across multiple views from a multi-camera rig or image sequences. The resulting solution is significantly more accurate than consumer-grade GNSS sensors and reaches accuracy levels closer to the traditional pipelines based on feature matching [261, 271].

Our approach, called OrienterNet, is a deep neural network that mimics the way humans orient themselves in their environment when looking at maps, i.e., by matching the metric 2D map with a mental map derived from visual observations [180, 215]. OrienterNet learns to compare visual and semantic data in an end-to-end manner, supervised by camera poses only. This yields accurate pose estimates by leveraging the high diversity of semantic classes exposed by OSM, from roads and buildings to objects like benches and trash cans. OrienterNet is also fast and highly interpretable. We train a single model that generalizes well to previously-unseen cities and across images taken by various cameras from diverse viewpoints – such as car-, bike- or head-mounted, pro or consumer cameras. Key to these capabilities is a new, large-scale training dataset of images crowd-sourced from cities around the world via the Mapillary platform.

Our experiments show that OrienterNet substantially outperforms previous works on localization in driving scenarios and vastly improves its accuracy in AR use cases when applied to data recorded by Aria glasses. We believe that our approach constitutes a significant step towards continuous, large scale, on-device localization for AR and robotics.

Map type	SfM SLAM	Satellite images	OpenStreetMap **(OrienterNet)**
What?	3D points +features	pixel intensity	polygons, lines, points
Explicit geometry?	3D	✗	2D
Visual appearance?	✓	✓	✗
Freely available	✗	✗	✓
Storage for 1 km^2	42 GB	75 MB	4.8 MB
Size reduction vs SfM	-	550×	8800×

Table 5.1.: ***Types of maps for visual localization.*** *Planimetric maps from OpenStreetMap consist of polygons and lines with metadata. They are publicly available for free and do not store sensitive appearance information, as opposed to satellite images and 3D maps built with SfM. They are also compact: a large area can be downloaded and stored on a mobile device. We show that they encode sufficient geometric information for accurate 3-DoF localization.*

5.2. Related work

We can localize an image in the world using several types of map representations: 3D maps built from ground images, 2D overhead satellite images, or simpler planimetric maps from OpenStreetMap. Table 5.1 summarizes their differences.

Mapping with ground-level images is the most common approach to date. Place recognition via image retrieval provides a coarse localization given a set of reference images [10, 101, 139, 322]. To estimate centimeter-accurate 6-DoF poses, algorithms based on feature matching require 3D maps [138, 187, 261, 271]. These are composed of sparse point clouds, which are commonly built with Structure-from-Motion (SfM) [2, 99, 176, 205, 278, 294] from sparse points matched across multiple views [30, 183, 253]. The pose of a new query image is estimated by a geometric solver [40, 114, 154] from correspondences with the map. While some works [306, 368] leverage additional sensor inputs, such as a coarse GNSS location, gravity direction, and camera height, recent localization systems are highly accurate and robust mostly thanks to learned features [81, 85, 245, 263, 330].

This however involves 3D maps with a large memory footprint as they store dense 3D point clouds with high-dimensional visual descriptors. There is also a high

risk of leaking personal data into the map. To mitigate this, some works attempt to compress the maps [45, 48, 187] or use privacy-preserving representations for the scene appearance [87, 210, 376] or geometry [297, 298]. These however either degrade the accuracy significantly or are easily reverted [234].

Localization with overhead imagery reduces the problem to estimating a 3-DoF pose by assuming that the world is mostly planar and that the gravity direction is often given by ubiquitous onboard inertial sensors. A large body of work focuses on cross-view ground-to-satellite localization. While more compact than 3D maps, satellite images are expensive to capture, generally not free, and still heavy to store at high resolution. Most approaches only estimate a coarse position through patch retrieval [133, 287, 289, 379]. In addition, works that estimate an orientation are not accurate [286, 288, 356], yielding errors of over several meters.

Other works rely on sensors that directly provide 3D metric information, such as 2D intensity maps from LiDAR [28, 188] or radar [26, 313]. They all perform template matching between 2D map and sensor overhead views, which is both accurate and robust, but require expensive specialized sensors, unsuitable for consumer AR applications. Our work shows how monocular visual priors can substitute such sensors to perform template matching from images only.

Planimetric maps discard any appearance and height information to retain only the 2D location, shape and type of map elements. OSM is a popular platform for such maps as it is free and available globally. Given a query area, its open API exposes a list of geographic features as polygons with metadata, including fine-grained semantic information with over a thousand different object types. Past works however design detectors for a single or few semantic classes, which lacks robustness. These include building outlines [12, 13, 55, 66, 77, 338, 341], road contours [97, 254] or intersections [189, 222, 358], lane markings [112, 225], street furniture [51, 351], or even text [129].

Recent works leverage more cues by computing richer representations from map tiles using end-to-end deep networks [259, 374]. They estimate only a coarse position as they retrieve map tiles with global image descriptors. In indoor scenes, floor plans are common planimetric maps used by existing works [130, 202]. They require height or visibility information that is typically not available for outdoor spaces. Our approach yields a significant step up in accuracy and robustness over all previous

works by combining the constraints of projective geometry with the expressivity of end-to-end learning, leveraging all semantic classes available in OSM.

5.3. Localizing single images in 2D maps

Problem formulation: In a typical localization scenario, we aim to estimate the absolute 6-DoF pose of an image in the world. Under realistic assumptions, we reduce this problem to estimating a 3-DoF pose $\boldsymbol{\xi} = (x, y, \theta)$ consisting of a location $(x, y) \in \mathbb{R}^2$ and heading angle $\theta \in (-\pi, \pi]$. Here we consider a topocentric coordinate system whose x-y-z axes correspond to the East-North-vertical directions.

First, we can easily assume to know the direction of the gravity, an information that humans naturally possess from their inner ear and that can be estimated by the inertial unit embedded in most devices. We also observe that our world is mostly planar and that the motion of people and objects in outdoor spaces is mostly restricted to 2D surface. The precise height of the camera can always be estimated as the distance to the ground in a local SLAM reconstruction.

Inputs: We consider an image $\mathbf{I}$ with known pinhole camera calibration. The image is rectified via a homography computed from the known gravity such that its roll and tilt are zero – its principal axis is then horizontal. We are also given a coarse location prior $\boldsymbol{\xi}_{\text{prior}}$. This can be a noisy GNSS position or a previous localization estimate and can be off by over 20 meters. This is a realistic assumption for a consumer-grade sensor in a multi-path environment like a urban canyon.

The map data is queried from OSM as a square area centered around $\boldsymbol{\xi}_{\text{prior}}$ and whose size depends on how noisy the prior is. The data consists of a collection of polygons, lines, and points, each of a given semantic class and whose coordinates are given in the same local reference frame.

Overview – Figure 5.2: OrienterNet consists of three modules: 1) The image-CNN extracts semantic features from the image and lifts them to an orthographic Bird's-Eye View (BEV) representation $\mathbf{T}$ by inferring the 3D structure of the scene. 2) The OSM map is encoded by the map-CNN into a neural map $\mathbf{F}$ that embeds

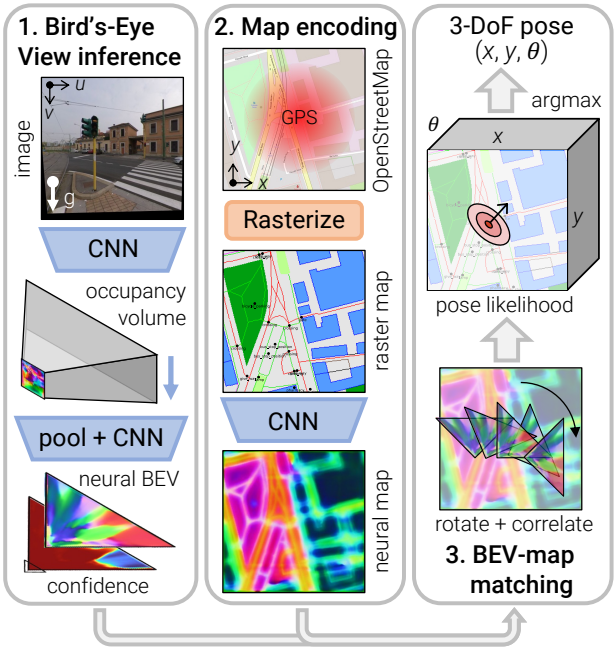


Figure 5.2.: ***OrienterNet architecture.*** *1) From an input image* $\mathbf{I}$ *that is gravity-aligned, we infer a mental map of the scene as a neural Bird's-Eye View (BEV)* $\mathbf{T}$ *with confidence* $\mathbf{C}$*. 2) From a coarse GNSS prior location* $\boldsymbol{\xi}_{prior}$*, we query OpenStreetMap and compute a neural map* $\mathbf{F}$*. 3) Matching the BEV against the map yields a probability volume* $\mathbf{P}$ *over 3-DoF camera poses. OrienterNet is trained end-to-end from pose supervision only.*

semantic and geometric information. 3) We estimate a probability distribution over camera poses $\boldsymbol{\xi}$ by exhaustively matching the BEV against the map.

5.3.1. Neural Bird's-Eye View inference

Overview: From a single image $\mathbf{I}$, we infer a BEV representation $\mathbf{T} \in \mathbb{R}^{L \times D \times N}$ distributed on a $L \times D$ grid aligned with the camera frustum and composed of N-dimensional features. Each feature on the grid is assigned a confidence, yielding a

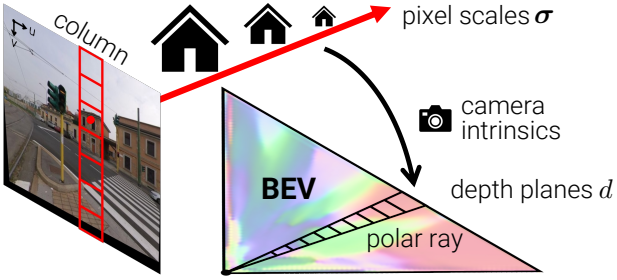


Figure 5.3.: ***OrienterNet predicts a pixel-wise distribution over scales*** *that are mapped to depths with the known camera calibration.*

matrix $\mathbf{C} \in [0,1]^{L \times D}$. This BEV is akin to a mental map that humans infer from their environment when self-localizing in an overhead map [180, 215].

Cross-modal matching between the image and the map requires extracting semantic information from visual cues. It has been shown that monocular depth estimation can rely on semantic cues [8] and that both tasks have a beneficial synergy [131, 159]. We thus rely on monocular inference to lift semantic features to the BEV space. Following past works that tackle semantic tasks [231, 248, 257], we obtain the neural BEV in two steps: i) we transfer image features to a polar representation by mapping image columns to polar rays, and ii) we resample the polar grid into a Cartesian grid (Fig. 5.3).

Polar representation: A CNN Φ_{image} first extracts a $U \times V$ feature map $\mathbf{X} \in \mathbb{R}^{U \times V \times N}$ from the image. We consider D depth planes sampled in front of the camera with a regular interval Δ, i.e. with values $\{i \cdot \Delta | i \in \{1 \dots D\}\}$. Since the image is gravity-aligned, each of the U columns in $\mathbf{X}$ corresponds to a vertical plane in the 3D space. We thus map each column to a ray in the $U \times D$ polar representation $\bar{\mathbf{X}} \in \mathbb{R}^{U \times D \times N}$. We do so by predicting, for each polar cell (u, d), a probability distribution $\boldsymbol{\alpha}_{u,d} \in [0,1]^V$ over the pixels in the corresponding image column:

$$\bar{\mathbf{X}}_{u,d} = \sum_v \boldsymbol{\alpha}_{u,d,v} \mathbf{X}_{u,v} \quad . \tag{5.1}$$

Instead of directly regressing the distribution $\boldsymbol{\alpha}$ over depths, we regress a distribution $\mathbf{S}$ over *scales* that are independent from the camera calibration parameters. The

scale is the ratio of object sizes in the 3D world and in the image [8] and is equal to the ratio of the focal length f and depth. We consider a set of S log-distributed scales

$$\boldsymbol{\sigma} = \left\{ \sigma_{\text{min}} \left(\sigma_{\text{max}} / \sigma_{\text{min}} \right)^{i/S} \, | i \in \{0 \dots S\} \right\} \quad . \tag{5.2}$$

Φ_{image} also predicts, for each pixel (u, v), a score vector $\mathbf{S}_{u,v} \in \mathbb{R}^S$ whose elements correspond to the scale bins $\boldsymbol{\sigma}$. We then obtain the distribution $\boldsymbol{\alpha}_{u,d}$ for each depth bin d as

$$\boldsymbol{\alpha}_{u,d,v} = \underset{v}{\text{softmax}} \left(\mathbf{S}_{u,v} \left[f / d \cdot \Delta \right] \right) \quad , \tag{5.3}$$

where $[\cdot]$ denotes the linear interpolation.

This formulation is equivalent to an attention mechanism from polar rays to image columns with scores resampled from linear depths to log scales. When the scale is ambiguous and difficult to infer, visual features are spread over multiple depths along the ray but still provide geometric constraints for well-localized map points [162]. Works tailored to driving scenarios [231, 248, 257] consider datasets captured by cameras with identical models and directly regress $\boldsymbol{\alpha}$. They therefore encode the focal length in the network weights, learning the mapping from object scale to depth. Differently, our formulation can generalize to arbitrary cameras at test time by assuming that the focal length is an input to the system.

BEV grid: We map the polar features to a Cartesian grid of size $L \times D$ via linear interpolation along the lateral direction from U polar rays to L columns spaced by the same interval Δ. The resulting feature grid is then processed by a small CNN Φ_{BEV} that outputs the neural BEV $\mathbf{T}$ and confidence $\mathbf{C}$.

5.3.2. Neural map encoding

We encode the planimetric map into a $W \times H$ neural map $\mathbf{F} \in \mathbb{R}^{W \times H \times N}$ that combines geometry and semantics.

Map data: OpenStreetMap [218] defines each map element as either a polygonal area, multi-segment lines, or a single point. Each element is annotated with a set of tags with standardized categories and labels according to a very rich hierarchy. We group elements into a smaller set of classes that we list in Tab. 5.2, resulting in 7 types

type	classes
areas	parking spot/lot, building, grass, playground, park, forest, water
lines	road, cycleway, pathway, busway, fence, wall, hedge, kerb, building outline, tree row
nodes	parking entrance, street lamp, junction, traffic signal, stop sign, give way sign, bus stop, stop area, crossing, gate, bollard, gas station, bicycle parking, charging station, shop, restaurant, bar, vending machine, pharmacy, tree, stone, ATM, toilets, water fountain, bench, waste basket, post box, artwork, recycling station, clock, fire hydrant, pole, street cabinet

Table 5.2.: ***List of map classes derived from OpenStreeMap data*** *and included in the map rasters.*

of areas, 10 types of lines, and 33 types of points (nodes). The accurate positioning of these elements provides geometric constraints necessary for localization, while their rich semantic diversity helps disambiguate different poses.

Preprocessing: We first rasterize the areas, lines, and points as a 3-channels image with a fixed ground sampling distance Δ, e.g. 50 cm/pixel. This representation is more informative and accurate than the naive rasterization of human-readable OSM tiles performed in previous works [259, 374].

Encoding: We associate each class with an N-dimensional embedding that is learned, yielding a $W \times H \times 3N$ feature map. It is then encoded into the neural map $\mathbf{F}$ by a CNN Φ_{map}, which extracts geometric features useful for localization. $\mathbf{F}$ is not normalized as we let Φ_{map} modulate its norm as importance weight in the matching. Examples in Fig. 5.4 reveal that $\mathbf{F}$ often looks like a distance field where we can clearly recognize distinctive features like corners or adjoining boundaries of buildings.

Φ_{map} also predicts a unary location prior $\mathbf{\Omega} \in \mathbb{R}^{W \times H}$ for each cell of the map. This score reflects how likely an image is to be taken at each location. We rarely expect images to be taken in, for example, rivers or buildings.

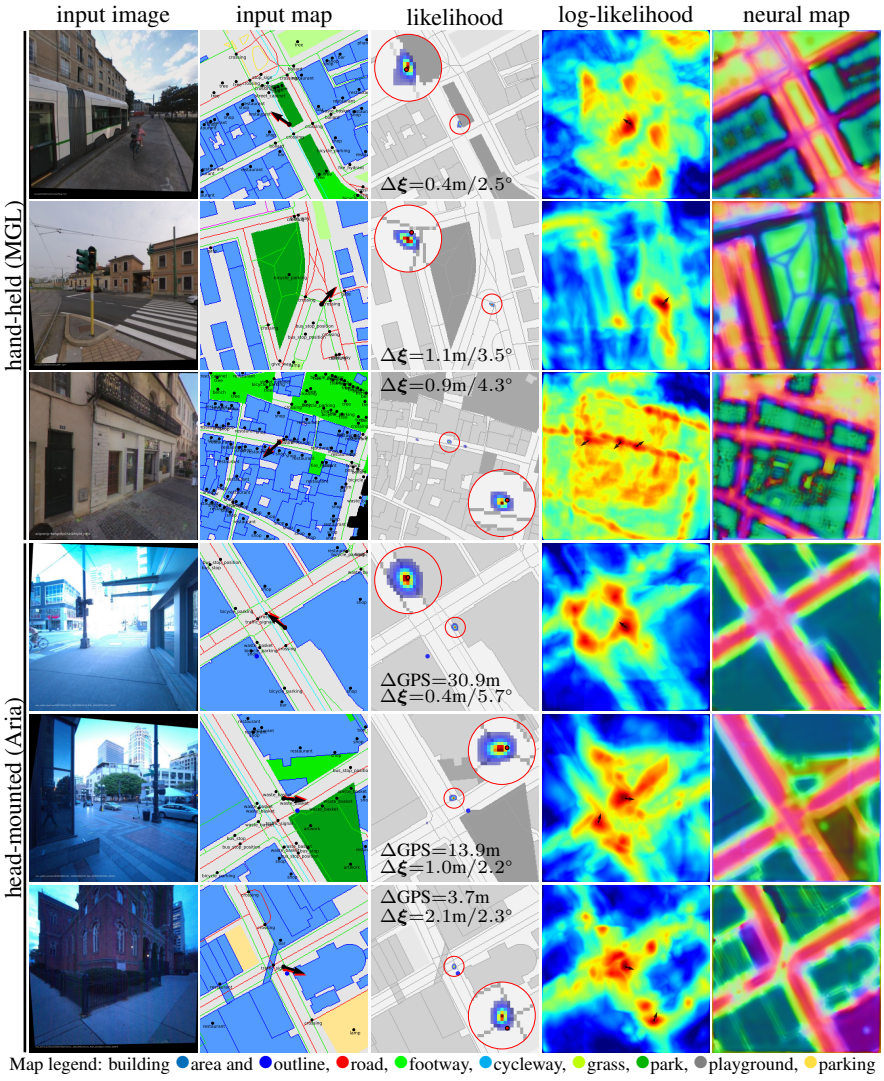


Figure 5.4.: ***OrienterNet generalizes well across datasets.*** *It handles different cameras, street-level viewpoints, and cities unseen during training. Overlayed on the input maps, the single-image predictions (➡) are close to the ground truth (➡) and more accurate than the noisy GNSS (dot ●). The model effectively leverages building corners and boundaries, crosswalks, sidewalks, road intersections, trees, and other elements.*

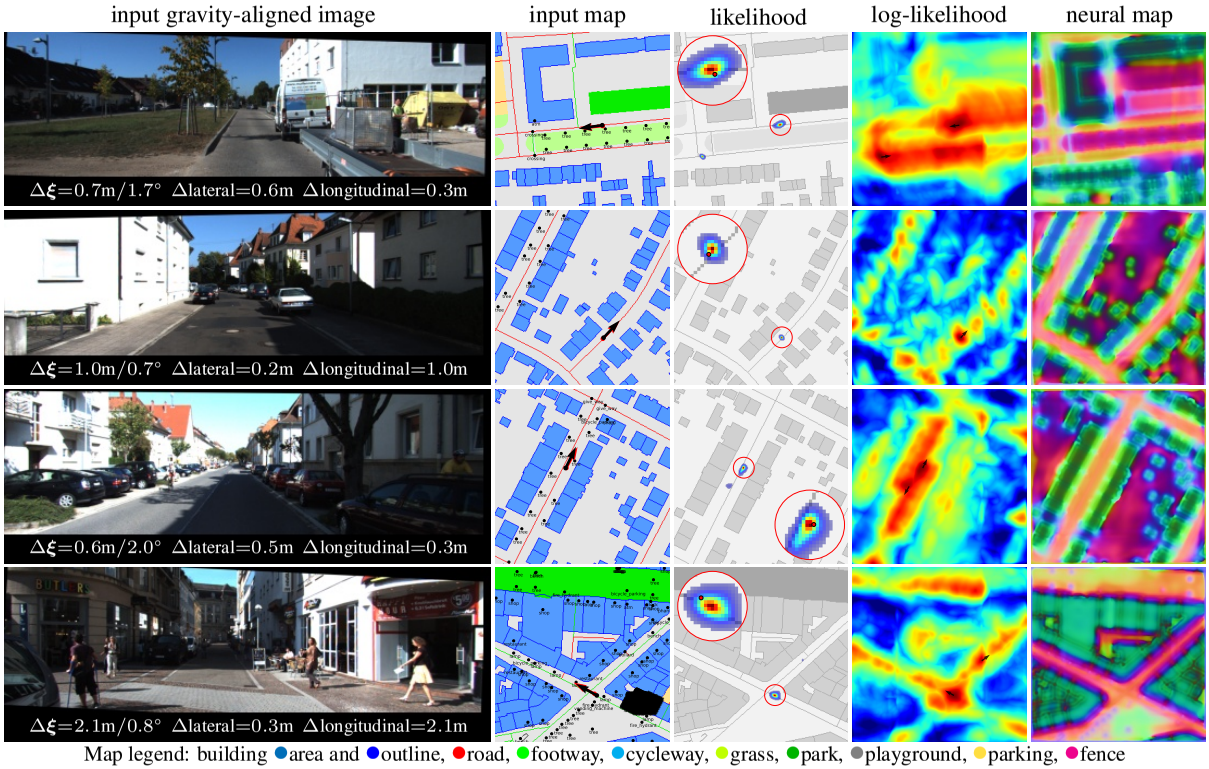


Figure 5.5.: ***OrienterNet can also handle images taken by car-mounted cameras.*** *Here images from the KITTI dataset have a very different resolution compared to the training images. OrienterNet nevertheless provides an accurate positioning. Likelihood maps can be multi-modal in scenes with repeated elements – we show the predicted orientations at local maxima with arrows.*

5.3.3. Pose estimation by template matching

Probability volume: We estimate a discrete probability distribution over camera poses $\boldsymbol{\xi}$. This is interpretable and fully captures the uncertainty of the estimation. As such, the distribution is multimodal in ambiguous scenarios. Figures 5.4 and 5.5 show various examples. This makes it easy to fuse the pose estimate with additional sensors like GNSS. Computing this volume is tractable because the pose space has been reduced to 3 dimensions. It is discretized into each map location and K rotations sampled at regular intervals.

This yields a $W \times H \times K$ probability volume $\mathbf{P}$ such that $P(\boldsymbol{\xi}|\mathbf{I}, \text{map}, \boldsymbol{\xi}_{\text{prior}}) = \mathbf{P}[\boldsymbol{\xi}]$. It is the combination of an image-map matching term $\mathbf{M}$ and the location prior $\boldsymbol{\Omega}$:

$$\mathbf{P} = \text{softmax}\left(\mathbf{M} + \boldsymbol{\Omega}\right) \quad . \tag{5.4}$$

$\mathbf{M}$ and $\boldsymbol{\Omega}$ represent image-conditioned and image-independent un-normalized log scores. $\boldsymbol{\Omega}$ is broadcasted along the rotation dimension and softmax normalizes the probability distribution.

Image-map matching: Exhaustively matching the neural map $\mathbf{F}$ and the BEV $\mathbf{T}$ yields a score volume $\mathbf{M}$. Each element is computed by correlating $\mathbf{F}$ with $\mathbf{T}$ transformed by the corresponding pose as

$$\mathbf{M}[\boldsymbol{\xi}] = \frac{1}{UZ} \sum_{\mathbf{p} \in (U \times Z)} \mathbf{F}[\boldsymbol{\xi}(\mathbf{p})]^\top \left(\mathbf{T} \odot \mathbf{C}\right)[\mathbf{p}] \quad , \tag{5.5}$$

where $\boldsymbol{\xi}(\mathbf{p})$ transforms a 2D point $\mathbf{p}$ from BEV to map coordinate frame. The confidence $\mathbf{C}$ masks the correlation to ignore some parts of the BEV space, such as occluded areas. This formulation benefits from an efficient implementation by rotating $\mathbf{T}$ K times and performing a single convolution as a batched multiplication in the Fourier domain [26, 28].

Pose inference: We estimate a single pose by maximum likelihood: $\boldsymbol{\xi}^* = \text{argmax}_{\boldsymbol{\xi}} P(\boldsymbol{\xi}|\mathbf{I}, \text{map}, \boldsymbol{\xi}_{\text{prior}})$. When the distribution is mostly unimodal, we can obtain a measure of uncertainty as the covariance of $\mathbf{P}$ around $\boldsymbol{\xi}^*$ [26].

5.4. Sequence and multi-camera localization

Single-image localization is ambiguous in locations that exhibit few distinctive semantic elements or repeated patterns. Such challenge can be disambiguated by accumulating additional cues over multiple views when their relative poses are known. These views can be either sequences of images with poses from SLAM or simultaneous views from a calibrated multi-camera rig. Figure 5.6 shows an example of such difficult scenario disambiguated by accumulating predictions over time. Different frames constrain the pose in different directions, e.g. before and after an intersection. Fusing longer sequences yields a higher accuracy (Fig. 5.7).

Let us denote $\boldsymbol{\xi}_i$ the unknown absolute pose of view i and $\hat{\boldsymbol{\xi}}_{ij}$ the known relative pose from view j to i. For an arbitrary reference view i, we express the joint likelihood over all single-view predictions as

$$P(\boldsymbol{\xi}_i|\{\mathbf{I}_j\}, \text{map}) = \prod_k P(\boldsymbol{\xi}_i \oplus \hat{\boldsymbol{\xi}}_{ij}|\mathbf{I}_j, \text{map}) \ , \tag{5.6}$$

where $\oplus$ denotes the pose composition operator. This is efficiently computed by warping each probability volume $\mathbf{P}_j$ to the reference frame i. We can also localize each image of a continuous stream via iterative warping and normalization, like in the classical Markov localization [41, 292].

5.5. Training a single strong model

Supervision: OrienterNet is trained in a supervised manner from pairs of single images and ground truth (GT) poses. The architecture is differentiable and all components are trained simultaneously by back-propagation. We simply maximize the log-likelihood of the ground truth pose $\boldsymbol{\xi}$: Loss $= -\log P(\boldsymbol{\xi}|\mathbf{I}, \text{map}, \boldsymbol{\xi}_{\text{prior}}) = -\log \mathbf{P}[\boldsymbol{\xi}]$. The tri-linear interpolation of $\mathbf{P}$ provides sub-pixel supervision.

Training dataset: We train a single model that generalizes to unseen locations with arbitrary kinds of images. We collect images from the Mapillary platform, which exposes the camera calibration, noisy GNSS measurement, and the 6-DoF pose in a global reference frame, obtained with a fusion of SfM and GNSS. The

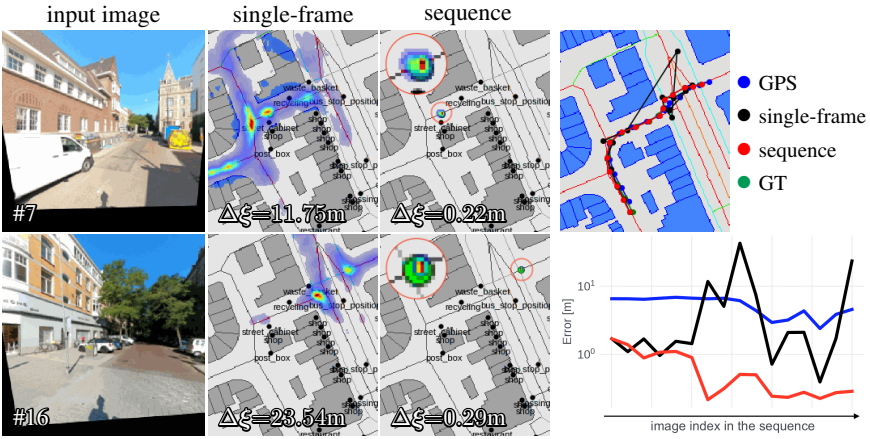


Figure 5.6.: ***Multi-frame fusion resolves ambiguities.*** *Semantic elements visible in a single image are often not sufficient to fully disambiguate the camera pose. Fusing the predictions over multiple frames collapses the multi-modal likelihood map to a single mode with high accuracy, yielding here a final error of less than 30cm.*

resulting *Mapillary Geo-Localization* (MGL) dataset includes 828k images from 12 cities in Europe and the US, captured by cameras that are handheld or mounted on cars or bikes, with GT poses and OSM data. We show the spatial distribution of the images in Fig. 5.8. Each city was divided into disjoint training and validation areas, resulting in 826k training and 2k validation images. Models trained on MGL generalize well to other datasets thanks to the diversity of cameras, locations, motions, and maps. All images are publicly available under a CC-BY-SA license via the Mapillary API. We believe that this dataset will significantly facilitate research on visual geo-localization.

Implementation: Φ_{image} and Φ_{map} are U-Nets with ResNet-101 and VGG-16 encoders. Φ_{BEV} has 4 residual blocks. We use $S{=}32$ scale bins, $K{=}512$ rotations. The BEV has size $L{\times}D{=}32{\times}32\,\text{m}$ with resolution $\Delta{=}50\,\text{cm}$. For training, we render maps $W{\times}H{=}128{\times}128\,\text{m}$ centered around points randomly sampled within 32 m of the GT pose. Localizing in such map takes 94 ms on an NVIDIA RTX 2080 GPU, with 37 ms for the BEV inference and 51 ms for the matching.

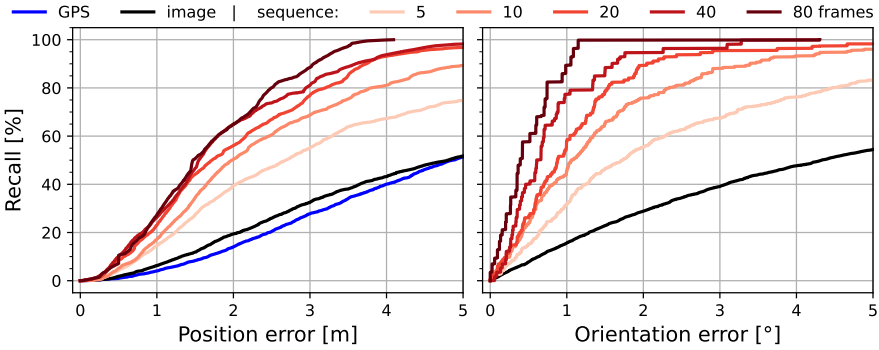


Figure 5.7.: ***With AR data, sequence localization boosts the recall****, which increases as we fuse information from additional frames.*

Model architecture	Position R@Xm			Orientation R@X°		
	1m	3m	5m	1°	3°	5°
Retrieval (a)	2.02	15.21	24.21	4.50	18.61	32.48
Refinement (b)	8.09	26.02	35.31	14.92	36.87	45.19
OrienterNet - planar (c)	14.28	44.59	56.08	20.43	50.34	64.30
OrienterNet - full	**15.78**	**47.75**	**58.98**	**22.14**	**52.56**	**66.32**

Table 5.3.: ***OrienterNet outperforms existing architectures****, which include: a) map tile retrieval by matching global embeddings [259, 356], b) featuremetric refinement [286] from an initial pose, and c) OrienterNet assuming a planar scene [286] instead of inferring monocular depth. We report the position and orientation recall (R).*

5.6. Experiments

We evaluate our single model for localization in the context of both driving and AR. Figures 5.4 and 5.5 shows qualitative examples, while Fig. 5.6 illustrates the effectiveness of multi-frame fusion. Our experiments show that: 1) OrienterNet is more effective than existing deep networks for localization with 2D maps; 2) Planimetric maps help localize more accurately than overhead satellite imagery; 3) OrienterNet is significantly more accurate than an embedded consumer-grade GNSS sensor when considering multiple views.

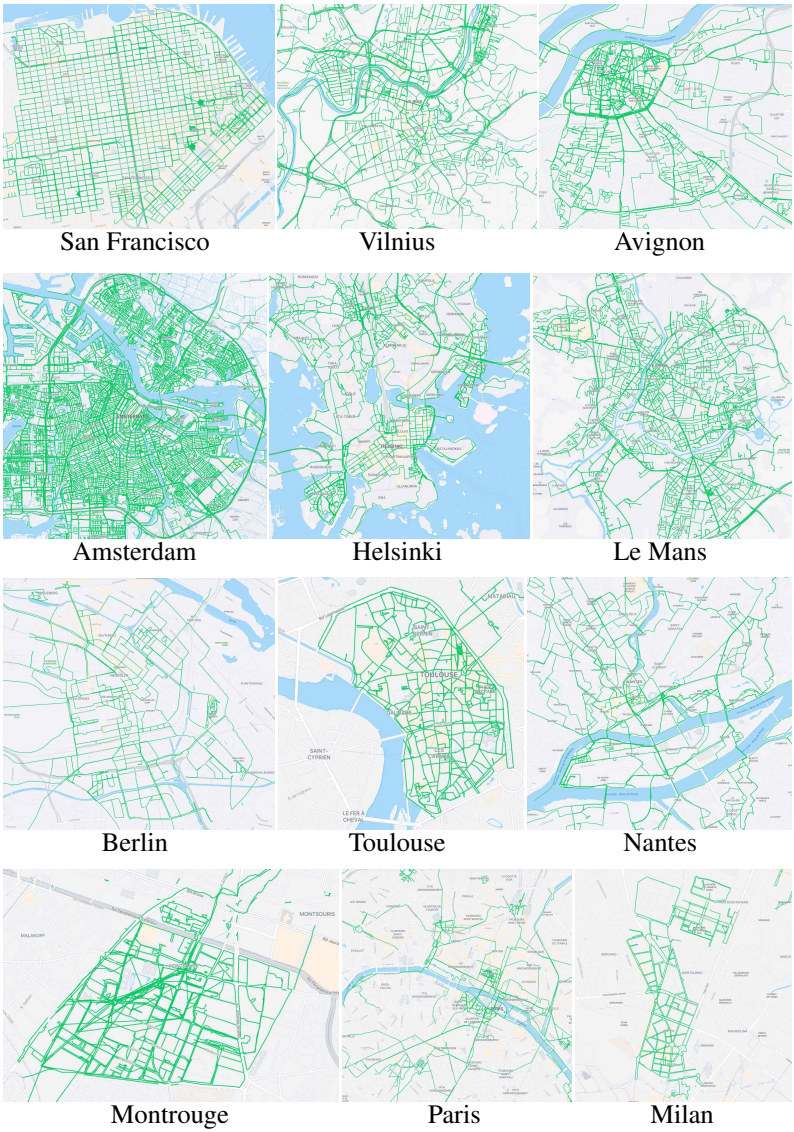


Figure 5.8.: ***Selected sequences of our MGL dataset*** *across 12 cities. Screenshots taken from the Mapillary platform browser.*

Map	Approach	Training dataset	Lateral R@Xm			Longitudinal R@Xm			Orientation R@X°		
			1m	3m	5m	1m	3m	5m	1°	3°	5°
Satellite	DSM [288]	KITTI	10.77	31.37	48.24	3.87	11.73	19.50	3.53	14.09	23.95
	VIGOR [379]	KITTI	17.38	48.20	70.79	4.07	12.52	20.14	-	-	-
	refinement [286]	KITTI	27.82	59.79	72.89	5.75	16.36	26.48	18.42	49.72	71.00
OpenStreetMap	retrieval [259, 356]	MGL	37.47	66.24	72.89	5.94	16.88	26.97	2.97	12.32	23.27
	refinement [286]	MGL	50.83	78.10	82.22	17.75	40.32	52.40	31.03	66.76	76.07
	OrienterNet (a)	MGL	53.51	88.85	94.47	26.25	59.84	70.76	34.26	73.51	89.45
	↳ **+ sequence** (b)	MGL	**79.71**	**97.44**	**98.67**	**55.21**	**95.27**	**99.51**	**77.87**	**97.76**	**100.**
	OrienterNet (c)	KITTI	51.26	84.77	91.81	22.39	46.79	57.81	20.41	52.24	73.53
	OrienterNet (d)	MGL+KITTI	65.91	92.76	96.54	33.07	65.18	75.15	35.72	77.49	91.51

Table 5.4.: ***Localization in driving scenarios with the KITTI dataset.*** *a) When trained on our MGL dataset, OrienterNet yields a higher localization recall than existing approaches based on both satellite imagery and OpenStreetMap, in terms of both orientation and lateral and longitudinal positional errors, b) Fusing predictions from sequences of 20 seconds boosts the recall. c) Training on KITTI outperforms other approaches trained on KITTI but is inferior to training on MGL. This demonstrates the excellent zero-shot capability of OrienterNet and the value of MGL. d) Pre-training on MGL and fine-tuning on KITTI achieves the best single-image performance.*

5.6.1. Understanding OrienterNet

Setup: We evaluate the design of OrienterNet on the validation split of our MGL dataset. This ensures an identical distribution of cameras, motions, viewing conditions, and visual features as the training set. We report recall of positions and rotation errors at the three thresholds 1/3/5m and 1/3/5°.

Comparing model architectures: We compare OrienterNet to alternative architectures trained on the same dataset: a) Map retrieval [356] replaces the BEV inference and matching by a correlation of the neural map and with a global image embedding. We predict a rotation by considering 4 different neural maps for the N-S-E-W directions. This formulation also regresses a probability volume and is trained identically to OrienterNet. It mimics the retrieval of densely-sampled map patches [259] but is significantly more efficient and practical. b) Featuremetric refinement [268, 286] updates an initial pose by warping a satellite view to the image assuming that the scene is planar, at a fixed height, and gravity-aligned. We replace the satellite view by an OSM map tile. This formulation requires an initial orientation (during both training and testing), which we sample within 45° of the ground truth. c) OrienterNet (planar) replaces the occupancy by warping the image features with a homography as in [286].

Analysis – Table 5.3: OrienterNet is significantly more accurate than all baselines at all position and rotation thresholds. a) Map retrieval disregards any knowledge of projective geometry and performs mere recognition without any geometric constraint. b) Featuremetric refinement converges to incorrect locations when the initial pose is inaccurate. c) Inferring the 3D geometry of the scene is more effective than assuming that it is planar. This justifies our design decisions.

Which map elements are most important? We study in Fig. 5.9 the impact of each type of map element on the final accuracy by dropping them from the input map. The classes with the largest impact are buildings and road, which are also the most common in areas covered by the training data.

Impact of the field of view (FoV): We study the impact of the FoV on the accuracy by cropping the images in the horizontal direction to varying degrees. Figure 5.10

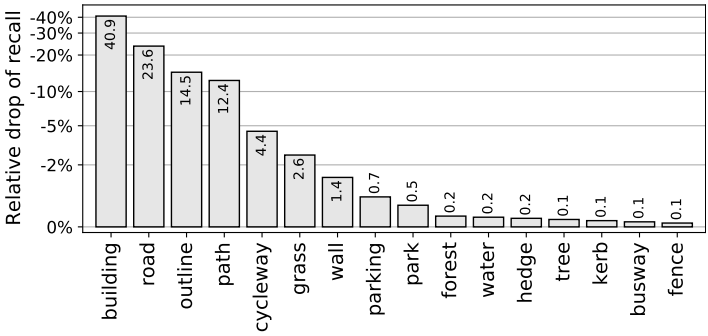


Figure 5.9.: ***Good semantics to localize.*** *Removing different elements from the map reveals how important they are for localization. Buildings, roads, footpaths, and cycleways are the most useful semantic classes, likely because they are also the most frequent.*

shows the results on the MGL validation set. Reducing the FoV decreases the accuracy proportionally – a 50% smaller FoV results in half of the original accuracy.

Model interpretability: We visualize in Fig. 5.11 multiple internal quantities that help us understand the predictions.

5.6.2. Application: robotics

Dataset: We consider the localization in driving scenarios with the KITTI dataset [102], following the closest existing setup [286]. To evaluate the zero-shot performance, we use their *Test2* split, which does not overlap with the KITTI and MGL training sets. Images are captured by cameras mounted on a car driving in urban and residential areas and have GT poses from high-accuracy GNSS. We augment the dataset with OSM maps.

Setup: We compute the position error along directions perpendicular (lateral) and parallel (longitudinal) to the viewing axis [286] since the pose is generally less constrained along the road. We report the recall at 1/3/5m and 1/3/5°. The original setup [286] assumes an accurate initial pose randomly sampled within ±20m and ±10° of the GT. OrienterNet does not require such initialization but only a coarse

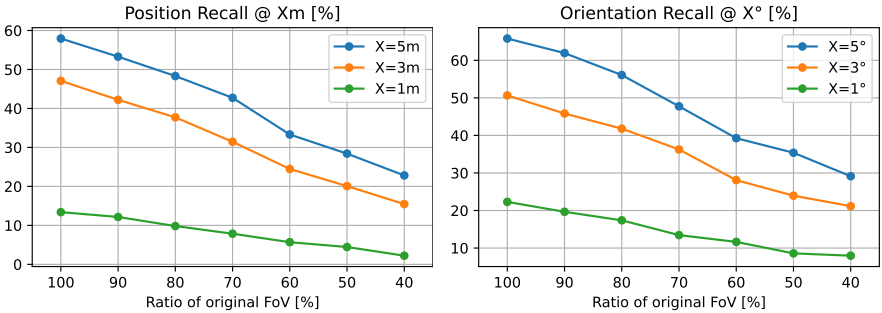


Figure 5.10.: ***Impact of the field of view (FoV)*** *on the localization recall with the MGL validation set. Decreasing the FoV directly impairs the accuracy as fewer map elements are visible in a single image.*

position-only prior. For fair comparisons, we nevertheless restrict the pose space to the same interval centered around the initial pose. We render 64×64 m map tiles and resize the images such that their focal length matches the median of MGL.

Baselines: We report approaches based on satellite maps and trained by [286] on KITTI. VIGOR [379] and DSM [288] both perform patch retrieval with global descriptors but respectively estimate an additional position offset or the orientation. We also evaluate the featuremetric refinement [268, 286] and baselines based on OSM maps, described in Sec. 5.6.1. As each scene is visited by a single trajectory, we cannot evaluate approaches based on 3D maps and image matching.

Results: Table 5.4 (a-b) shows that OrienterNet outperforms all existing approaches based on both satellite and OSM maps, in all metrics. OrienterNet exhibits remarkable zero-shot capabilities as it outperforms approaches trained on KITTI itself. The evaluation also demonstrates that planimetric maps yield better localization, as retrieval and refinement approaches based on them outperform those based on satellite images. The recall at 3m/3°is saturated to over 95% by fusing the predictions from sequences of only 20 seconds.

Generalization: Table 5.4 (c-d) shows that training OrienterNet solely on KITTI results in overfitting, as the dataset is too small to learn rich semantic representations. Our larger MGL dataset alleviates this issue and enables cross-modal learning with

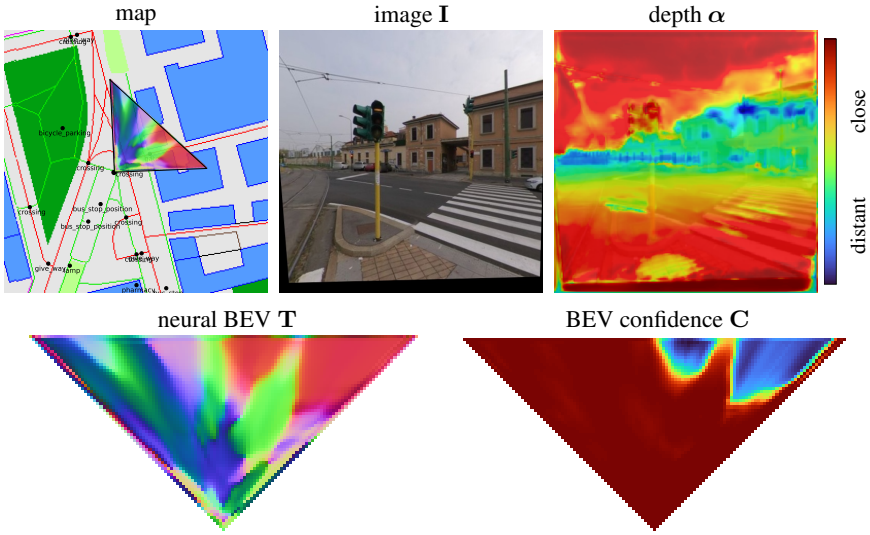


Figure 5.11.: ***End-to-end but interpretable.*** *From only pose supervision, OrienterNet learns to infer the 3D geometry of the scene via the depth planes* α *and the 2D occupancy via the confidence* $\mathbf{C}$*.*

rich semantic classes. Pre-training on MGL and fine-tuning on KITTI yields the best performance.

5.6.3. Application: augmented reality

We now consider the localization of head-mounted devices for augmented reality (AR). We show that OrienterNet is more accurate than a typical embedded GNSS sensor.

Dataset: There is no public benchmark that provides geo-aligned GT poses for images captured with AR devices in diverse outdoor spaces. We thus recorded our own dataset with Aria devices [92]. It exhibits patterns typical of AR with noisy consumer-grade sensors and pedestrian viewpoints and motions. We include two locations: i) Seattle (Downtown, Pike Place Market, Westlake), with high-rise

City	Setup	Approach	Position R@Xm			Orientation R@X°		
			1m	3m	5m	1°	3°	5°
Seattle	single	GPS	1.25	8.82	18.44	-	-	-
		retrieval [259, 356]	0.88	3.81	5.95	2.83	8.36	12.96
		OrienterNet	3.39	14.49	23.92	6.83	20.39	30.89
	multi	GPS	1.76	9.2	20.48	4.18	11.01	23.36
		OrienterNet	**21.88**	**61.26**	**72.92**	**33.86**	**72.41**	**83.93**
Detroit	single	GPS	3.96	27.75	51.33	-	-	-
		retrieval [259, 356]	3.31	19.83	36.76	6.48	18.40	28.88
		OrienterNet	6.26	32.41	51.76	15.53	39.06	54.41
	multi	GPS	4.09	31.36	53.41	13.48	37.84	55.24
		OrienterNet	**17.18**	**68.77**	**89.26**	**44.85**	**88.04**	**96.04**

Table 5.5.: ***Localization of head-mounted devices for AR.*** *With data from Aria glasses, OrienterNet outperforms the map retrieval baseline and the embedded GNSS sensor in both single- and multi-frame settings, in both cities. Multi-frame fusion does not filter out the high noise of the GNSS but strongly benefits our approach.*

buildings, and ii) Detroit (Greektown, Grand Circus Park), with city parks and lower buildings. We record several image sequences per city, all roughly following the same loop around multiple blocks. Each device is equipped with a consumer-grade GNSS sensor, IMUs, grayscale SLAM cameras, and a front-facing RGB camera, which we undistort to a pinhole model.

We obtained relative poses and gravity direction from an offline proprietary visual-inertial SLAM system. We then computed pseudo-GT global poses by jointly optimizing all sequences based on GNSS, SLAM constraints, and predictions of OrienterNet. We selected query frames every 3 meters, resulting in 2153 frames for Seattle and 2725 frames for Detroit. For each evaluation example, the map tile is centered around the noisy GNSS measurement. Because of large differences in GNSS accuracy due to urban canyons, we constrain the predictions within 64 m of the measurement for Seattle and 24 m for Detroit.

Single-frame localization – Table 5.5: OrienterNet is consistently more accurate than the GNSS, which is extremely noisy in urban canyons like Seattle because of multi-path effects. The performance is however significantly lower than with

driving data (Sec. 5.6.2), which highlights the difficulty of AR-like conditions and the need for further research.

Multi-frame: We now fuse multiple GNSS signals or predictions of OrienterNet over the same temporal interval of 10 consecutive keyframes, using imperfect relative poses from SLAM. The fusion more than doubles the accuracy of OrienterNet but marginally benefits the GNSS sensor because of its high, biased noise, especially in Seattle.

Comparison to feature matching: Algorithms based on 3D SfM maps require mapping images, whose quality and density have a large impact on the localization accuracy. Differently, OrienterNet can localize in areas not covered by such images as long as OSM data is available. This makes any fair comparison difficult.

5.7. Summary and outlook

Summary: OrienterNet is the first deep neural network that can localize an image with sub-meter accuracy within the same 2D planimetric maps that humans use. OrienterNet mimics the way humans orient themselves in their environment by matching the input map with a mental map derived from visual observations. Compared to large and expensive 3D maps that machines have so far relied on, such 2D maps are extremely compact and thus finally enable on-device localization within large environments. OrienterNet is based on globally and freely available maps from OpenStreetMap and can be used by anyone to localize anywhere in the world.

We contribute a large, crowd-sourced training dataset that helps the model generalize well across both driving and AR datasets. OrienterNet significantly improves over existing approaches for 3-DoF localization, pushing the state of the art by a large margin. This opens up exciting prospects for deploying power-efficient robots and AR devices that know where they are without costly cloud infrastructures.

Impact and limitations: OrienterNet sparked interest in the open-source intelligence community for its ability to estimate the location of any Internet image that has a coarse location prior. Its accuracy is however far below the one of top

players of GeoGuessr. Figures 5.12 and 5.13 show some failure cases. Localizing an image or a sequence is challenging when the environments lacks distinctive elements or when they are not registered in the map. The latter can occur when the environment changes between the times at which the image is taken and the map is last updated. This also occurs because the labeling of OSM is not homogeneous and varies in different regions of the world, especially in areas with a low density of population. For example, trees are not consistently always label in OSM. The spatial accuracy of OSM is also unknown and likely poor in some areas. Overall, this makes OrienterNet hardly reliable in practical applications. To bridge this gap, we introduce in Chapter 6 an approach to estimate a new kind of map that is optimal for visual localization, given only raw imagery.

In this part of the thesis, we have assumed that the gravity direction is known. Our experiment assume a perfect gravity estimate and do not evaluate the impact of an inaccurate gravity. This is mostly true when the gravity is given by inertial measurements fused across time. When those are not available, one can use a deep neural network to estimate the gravity [127, 142, 182, 337], but with a lower accuracy.

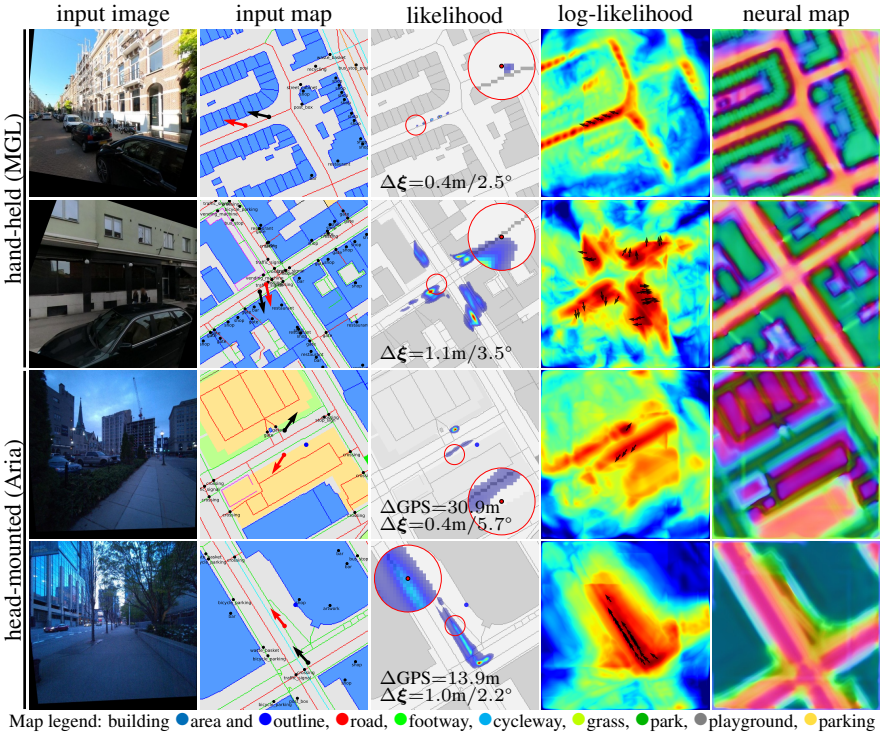


Figure 5.12.: ***Failure cases of single-image localization (1/2).*** *Localizing a single image often fails when the environment lacks distinctive elements, when they do not appear in the map, or when such elements are repeated, making the pose ambiguous. Since OSM is crowd-sourced, the level of detail of the map is not consistent and widely varies. For example, trees are registered in some cities but not in others (last row).*

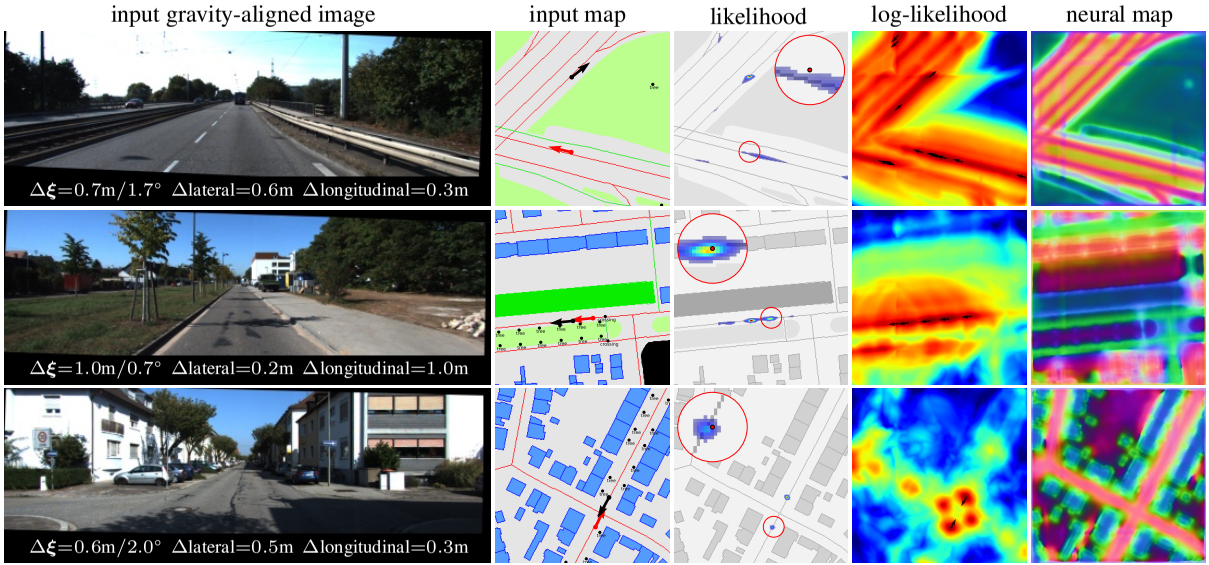


Figure 5.13.: ***Failure cases of single-image localization (2/2).*** *1) Highways are particularly challenging because they generally lack distinctive map elements, as traffic signs and poles often do not appear in OpenStreetMap. 2) Other scenarios can also be ambiguous, such as this row of trees. 3) Symmetric intersections without distinctive features are also challenging.*

CHAPTER 6

Self-Supervised Neural Maps for Visual Positioning and Semantic Understanding

Semantic 2D maps are commonly used by humans and machines for navigation purposes, whether it's walking or driving. However, these maps have limitations: they lack detail, often contain inaccuracies, and are difficult to create and maintain, especially in an automated fashion. Can we use *raw imagery* to automatically create *better maps* that can be easily interpreted by both humans and machines? We introduce SNAP, a deep network that learns rich *neural* 2D maps from ground-level and overhead images. We train our model to align neural maps estimated from different inputs, supervised only with camera poses over tens of millions of StreetView images. SNAP can resolve the location of challenging image queries beyond the reach of traditional methods, outperforming the state of the art in localization by a large margin. Moreover, our neural maps encode not only geometry and appearance but also high-level semantics, discovered without explicit supervision. This enables effective pre-training for data-efficient semantic scene understanding, with the potential to unlock cost-efficient creation of more detailed maps.

6.1. Introduction

Semantic 2D maps such as Google Maps are ubiquitous in our daily lives, used by billions of people. They offer compact, yet easily interpretable representations of

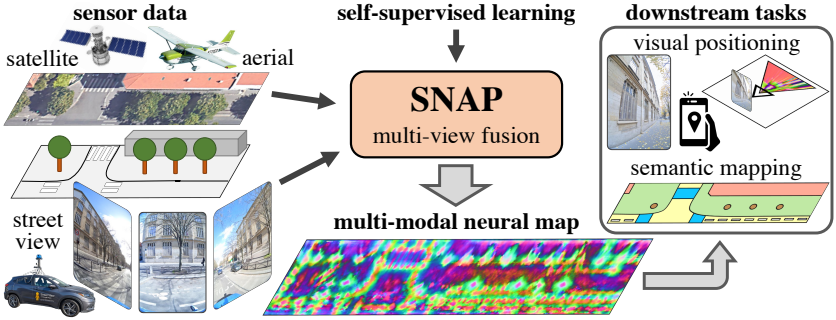


Figure 6.1.: *We learn* neural 2D maps *from multi-modal imagery using camera poses. SNAP outperforms the state of the art in visual positioning, and by solving localization as a proxy task learns easily interpretable, high-level semantics through self-supervision alone, without any semantic cues.*

the world from a bird's-eye view, allowing us to effectively navigate large outdoor environments by foot or vehicle. By contrast, machines position themselves in the real world through computer vision, which remains dominated by structure-based approaches [135, 187, 228, 261, 270, 278] relying on basic hand-crafted [11, 183] or learned [81, 177, 203, 245, 263, 330, 364] primitives, such as points or lines. These approaches build 3D maps with Structure-from-Motion (SfM) and then localize query images via 2D-3D registration. Their complexity and many components (feature extraction and matching, bundle adjustment, pose refinement, etc.) make it difficult to tune [143] or update [84] them, and to learn high-level priors end-to-end [32, 38, 268]. They are also costly to store and generally not reusable for other applications.

OrienterNet [264], as introduced in Chapter 5, instead learn planar, neural representations from the same 2D semantic maps that humans use. These maps encode scene geometry and semantics and can be used for visual positioning with sub-meter accuracy. This approach is however limited to a few semantic classes, and the maps it is based on can be inaccurate, costly to obtain, and difficult to maintain.

We argue that maps are most useful for figuring out where we are when they are *abstract* enough to be robust to temporal changes, yet preserve enough *geometric and semantic information* to yield high-quality correspondences with the physical

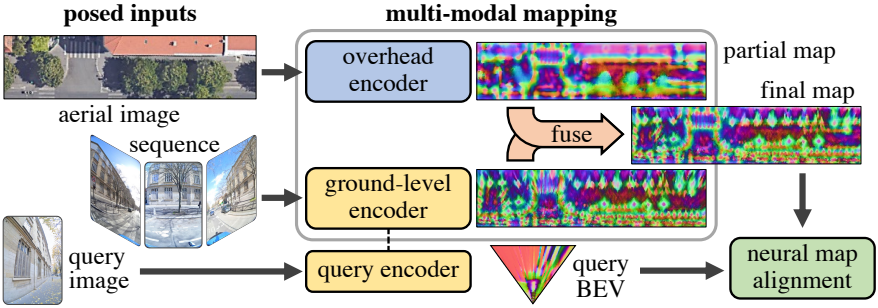


Figure 6.2.: ***Training architecture.*** *We feed overhead and ground-level imagery to per-stream encoders (Sec. 6.2.1) to produce 2D bird's-eye-view neural maps, fused via cell-wise max-pooling (Sec. 6.2.2). We also extract a 'query' neural map from a single ground-level image with the same ground-level encoder. Given known poses, we train SNAP by simply registering 'query' and 'scene' maps (Sec. 6.3).*

world. Our work, SNAP, shows that, by learning 2D neural maps for localization, meaningful semantics emerge without explicitly supervising them. These semantics improve positioning accuracy and also make our maps usable for other tasks (Fig. 6.1).

SNAP leverages the complementary strengths of different input modalities, like ground-level and overhead imagery, by fusing them into a single 2D neural map (Fig. 6.2). It can flexibly and efficiently integrate arbitrary combinations of data captured at different points in time, which is key to continuously update maps in a changing world. We train it end-to-end to estimate the pose of a query image relative to the mapping images, by simply aligning their neural maps. This kind of contrastive learning requires only sensor poses, which can be easily obtained with photogrammetry [123, 153]. We train and evaluate SNAP on a dataset with 50M StreetView images* from 5 continents, *orders-of-magnitude* larger and more diverse than comparable academic benchmarks.

Despite training only for a positioning objective, we observe that our neural maps learn easily-interpretable, high-level semantics without the need for explicit semantic cues (Fig. 6.8), and demonstrate that they provide an effective pretraining for semantic understanding tasks by fine-tuning them on little labeled data. This can

* Analytical use of StreetView imagery was done with special permission from Google.

potentially unlock cost-efficient creation of more detailed and richer maps, readable by humans and machines alike, while providing state-of-the-art visual positioning.

Our main contributions are as follows. (i) We introduce a simple and lightweight encoder to estimate bird's-eye view maps from ground-level imagery, combining principles from multi-view geometry with strong monocular cues. (ii) We fuse different imaging modalities to integrate and benefit from complementary cues. (iii) We show how to train our model by aligning neural maps in a contrastive learning framework, using RANSAC to mine hard negatives. (iv) We outperform the state of the art on visual positioning and register image queries beyond the reach of traditional methods (Fig. 6.8). (v) We demonstrate that high-level semantics emerge by learning to align neural maps, without any explicit supervision, and fine-tune them on semantic understanding with few labels (Fig. 6.17).

6.2. Mapping the world with neural maps

We now formalize neural maps, and describe a neural network architecture to infer them from raw sensor data. Our goal is to infer a more generic neural representation that can encode both the geometry, semantics, and appearance of a given point in the 2D world.

Problem formulation: For a 3D scene, such as a large outdoor environment, we consider a local, 3D Cartesian coordinate system such that the z axis points upwards along the gravity direction. A neural map $\mathbf{M}$ is defined over a regular grid that partitions the xy plane into $I \times J$ square cells of size Δ. Each cell (i, j) is associated with a D-dimensional feature $\mathbf{M}_{ij} \in \mathbb{R}^D$. To infer such neural map, we leverage large quantities of raw imagery captured by diverse cameras.

Input modalities: Ground-level images are captured by cameras mounted on StreetView cars or backpacks [62]. They are often part of a sequence of multi-camera frames. As such, they are very unevenly distributed throughout space. Each image offers a high resolution view of a small area, mainly limited by the occlusion of static or dynamic objects like buildings or vehicles. On the other hand, overhead images are captured by cameras mounted on planes or satellites. These images benefit from high spatial coverage at a uniform but low resolution. Their visibility is

mostly affected by vertical occluders like trees. Ground-level and overhead images capture different aspects of the environment and are thus complementary.

Assumptions: All images of either modality are calibrated and registered with respect to the map coordinate system. Each image n follows a projection function $\Pi_n : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ that maps a 3D point in the world to a 2D point on the image plane. Π_n combines the camera pose ${}_w\mathbf{T}_n \in \mathrm{SE}(3)$ and the camera calibration, including lens distortions. Overhead images are ortho-rectified, such that world points along the z axis project onto the same pixel coordinate. As this process relies on a coarse digital surface model [93], fine details like poles are not rectified and may result in artifacts, which SNAP can however learn to account for.

6.2.1. Fusing multi-modal representations

Each location in the world is observed by an arbitrary number of images for each modality, captured at arbitrary points in time. We thus follow a late-fusion strategy that first encodes each modality separately and only finally fuses them (Fig. 6.2). This can flexibly adapt to the available inputs and efficiently handle arbitrary spatial distributions of data.

Encoding: We design two encoders that each combine a subset of observations n into a single-modality neural map $\mathbf{M}^n$ defined over the same grid as $\mathbf{M}$. Φ_{OV} encodes a single tile of overhead orthoimagery, while Φ_{SV} encodes a single image or multiple covisible ground-level StreetView images, e.g., a multi-view sequence. To best resolve the 3D information from perspective shots at arbitrary viewpoints, Φ_{SV} leverages both multi-view observations and monocular cues. We describe its architecture in detail in Sec. 6.2.2. Φ_{OV}, on the other hand, is a simple U-Net-style CNN [250] that computes a feature for each pixel of the overhead orthoimage, which is then resampled into the grid.

Fusion: We obtain the final neural map by fusing the set of encoded maps $\{\mathbf{M}^n\}$ using a cell-wise max-pooling operation, i.e., $\mathbf{M}_{ij} = \max_n \mathbf{M}^n_{ij} \; \forall \; (i, j) \in I \times J$. This can combine maps with different spatial extents, which is essential to scale to large areas. The *max* aggregation picks the best estimate among all inputs for each

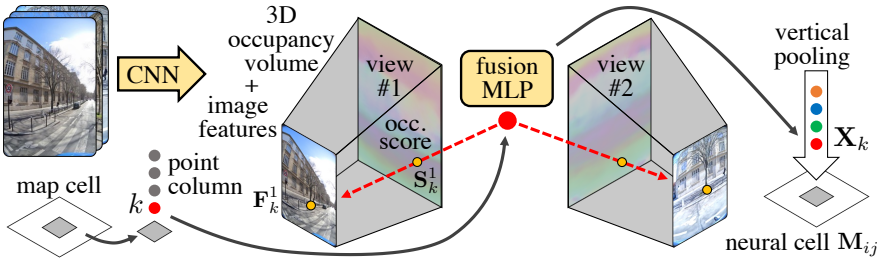


Figure 6.3.: ***Ground-level encoder: combining multi-view geometry and monocular priors.*** *We use a CNN to predict pixel-wise features and a monocular occupancy volume, separately for each view. We then interpolate them over a column of 3D points (at predefined heights), for each 2D cell. Finally, a simple MLP combines them into features $\mathbf{X}_k$ that are pooled along the column, into a neural cell.*

feature channel and thus handles partial observations, such as when the road surface cannot be resolved in overhead images because it is occluded by trees.

6.2.2. Ground-level image encoder

We design a single module, Φ_{SV}, that can arbitrarily encode one or multiple images, ordered or not. Φ_{SV} first fuses the image data into 3D space and later projects it vertically into the map plane (Fig. 6.3). This design can handle arbitrary ground geometries and accurately resolve the 2D location of overhanging 3D structures, like street lights. The 3D fusion leverages both multi-view geometry and strong monocular cues learned end-to-end. Φ_{SV} can thus resolve objects that are observed by a single image, while maximizing accuracy when multiple observations are available.

Monocular inference: We consider an unordered set of N images $\{\mathbf{I}^n\}$, $N \geq 1$. Each image n is encoded independently by a CNN $\Phi_{\mathbf{I}}$ into a C-dimensional feature image $\mathbf{F}^n \in \mathbb{R}^{H \times W \times C}$. $\Phi_{\mathbf{I}}$ also estimates a pixel-wise depth $\mathbf{S}^n \in \mathbb{R}^{H \times W \times D}$ as a score over D depth planes along the ray of each pixel. $\mathbf{S}^n$ is similar to a frustum-aligned occupancy volume [231, 264] but contains unnormalized logits of a depth distribution. Instead of regressing a single value, this encodes the full depth uncertainty along the ray and thus allows $\Phi_{\mathbf{I}}$ to provide meaningful multi-modal

estimates. We distribute the depth planes uniformly in log space to correlate with the uncertainty of monocular depth estimation [8, 264].

Multi-view fusion: To fuse information in 3D, we define K horizontal planes at heights $\{z_k\}$, which are uniformly distributed within a range of interest defined with respect to the height of the camera [173, 284], e.g., from 4 m below to 8 m above. For a 2D map cell $(i, j) \in I \times J$, we consider its center point (x, y) and a column of 3D points $\{\mathbf{P}_k = (x, y, z_k)\}$. For each 3D point k, we define the subset of views that best observe it as $\mathcal{N}_k \subseteq \{1 \dots N\}$, e.g., those that are closest spatially. We project the point to each of these views, obtain a 2D observation $\mathbf{p}_k^n = \Pi_n(\mathbf{P}_k)$, and sample the corresponding feature image with bi-linear interpolation: $\mathbf{F}_k^n = \mathbf{F}^n[\mathbf{p}_k^n]$. Given the depth d_k^n of $\mathbf{P}_k$ in the corresponding view, we also tri-linearly interpolate a score from the depth prior: $\mathbf{S}_k^n = \mathbf{S}^n[\mathbf{p}_k^n, d_k^n]$. Intuitively, $\mathbf{S}_k^n$ is low if the 3D point is in free space or is occluded in view n. Following common practice in learned multi-view stereo [347, 362], we then compute feature consistency statistics, as mean and variance $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k) \in \mathbb{R}^C$, weighted by the depth priors:

$$\boldsymbol{\mu}_k = \sum_{n \in \mathcal{N}_k} w_k^n \, \mathbf{F}_k^n \quad \text{and} \quad \boldsymbol{\sigma}_k = \sum_{n \in \mathcal{N}_k} w_k^n \, (\mathbf{F}_k^n - \boldsymbol{\mu}_k)^2 \quad \text{with} \quad w_k^n = \operatorname*{softmax}_{n \in \mathcal{N}_k} \mathbf{S}_k^n \ . \tag{6.1}$$

A Multi-Layer Perceptron (MLP) fuses this information into a feature $\mathbf{X}_k$, which is finally pooled across all points in the column, resulting in a neural map cell $\mathbf{M}_{ij}$:

$$\mathbf{M}_{ij} = \max_k \mathbf{X}_k \quad \text{with} \quad \mathbf{X}_k = \text{MLP}\left(\left[\boldsymbol{\mu}_k, \, \boldsymbol{\sigma}_k, \, \max_{n \in \mathcal{N}_k} \mathbf{S}_k^n\right]\right) \ . \tag{6.2}$$

Adding the maximum depth score differentiates free and occupied space when the point is observed by a single image. This makes it possible to use the same model for single images and sequences.

By tightly combining 3D geometry and representation learning, our approach leverages both monocular priors and multi-view information, while past research on 2D mapping or 3D reconstruction typically relies on only one of the two. Compared to expensive Transformers [331] or 3D CNNs [110], we show that a simpler, lightweight MLP is effective at fusing multi-view information, inspired by [276]. Compared to top-down 2D CNNs that squash the vertical dimension [115, 249], this MLP is more expressive and makes our neural maps equivariant to 2D translations and rotations and invariant to translations along the vertical axis. Overall, this

simple design enables scaling to very large scenes, which is critical to provide hard negatives for contrastive learning and ultimately learn rich semantics.

6.3. Learning from pose supervision

Alignment as contrastive learning: We want neural maps to encode high-level semantic information about the environment. Given recent advances in self-supervised learning [50, 219], we hypothesize that this can emerge from learning distinctive features that distinguish one location from another and that are invariant to viewpoint and temporal appearance changes. Intuitively, *good maps help us identify where we are*. More generally, good maps are such that we can unambiguously align them when inferred from partial inputs. Consider neural maps $\mathbf{M}^Q$ and $\mathbf{M}^R$ obtained from two disjoint subsets of inputs, the query Q and the reference R. In camera pose estimation, Q corresponds to a single ground-level image and R to a sequence of images with an aerial tile. Because our encoder is flexible, we can use the same shared model to encode Q and R (Fig. 6.2). $\mathbf{M}^Q$ is defined over a grid $\mathbf{G}^Q \in \mathbb{R}^{I \times J \times 2}$ in a local coordinate frame, e.g., aligned with the query camera, where $\mathbf{G}^Q_{ij}$ is the center point of cell (i, j), while $\mathbf{M}^R$ is defined in the world frame.

We define a score function $E(\mathbf{T}; \mathbf{M}^Q, \mathbf{M}^R) : \mathrm{SE}(2) \rightarrow \mathbb{R}$ that evaluates the consistency between $\mathbf{M}^Q$ and $\mathbf{M}^R$ given an estimate of their 3-DoF relative pose ${}_R\mathbf{T}_Q \in \mathrm{SE}(2)$. To distinguish the ground-truth pose ${}_R\mathbf{T}^*_Q$ from K other, incorrect poses $\{{}_R\mathbf{T}^k_Q\}$, we want to increase $E({}_R\mathbf{T}^*_Q)$ and decrease $E({}_R\mathbf{T}^k_Q)$ (omitting $\mathbf{M}^Q$ and $\mathbf{M}^R$ for brevity). This corresponds to a contrastive learning problem, for which we minimize the InfoNCE loss [217]

$$\mathrm{Loss}\left(\mathbf{M}^Q, \mathbf{M}^R\right) = -\log \frac{\exp\left(E\left({}_R\mathbf{T}^*_Q\right)/\tau\right)}{\sum_{k \in \{*, 1 \ldots K\}} \exp\left(E\left({}_R\mathbf{T}^k_Q\right)/\tau\right)}, \tag{6.3}$$

where τ is a learnable temperature parameter. Neural maps are trained end-to-end and require only relative poses ${}_R\mathbf{T}^*_Q$, which can be easily obtained at a large scale using photogrammetry [123, 153].

Featuremetric pose scoring: A linear layer projects each neural map $\mathbf{M}$ to a lower-dimensional, L2-normalized map $\bar{\mathbf{M}}$. This creates an information bottleneck

that encourages compact features. The score E evaluates the consistency of two neural maps as the similarity of each cell after warping:

$$E({}_R\mathbf{T}_Q) = \frac{1}{IJ} \sum_{(i,j)\in I\times J} \max\left(\bar{\mathbf{M}}_{ij}^{Q\top} \bar{\mathbf{M}}^R \left[{}_R\mathbf{T}_Q \cdot \mathbf{G}_{ij}^Q\right], 0\right) \quad , \tag{6.4}$$

where ${}_R\mathbf{T}_Q$ transforms a grid point from coordinate frames Q to R and $[\cdot]$ interpolates the map at this location. $\max$ clips negative scores to zero to reduce the impact of outliers, as in robust optimization.

Negative sampling: A critical and well-studied aspect of contrastive learning is the selection of negative samples [119, 316, 355]. Hard negatives should be high-likelihood but incorrect predictions, so as to push the probability mass to the ground truth. Random poses can be easily distinguished and exhaustive voting in the 3-DoF pose space is computationally infeasible at high resolution [28, 94, 264]. Instead, we use RANSAC [95] to sample poses that are consistent with the predicted features. We sample pairs of 2D-2D correspondences between all cells of both neural maps and solve for the relative pose using the Kabsch algorithm [146]. Inspired by PROSAC [67], we sample a correspondence between cells (i, j) and (k, l) based on its feature similarity with probability $P_{ijkl} = \underset{ijkl}{\text{softmax}}\left(\bar{\mathbf{M}}_{ij}^{Q\top} \bar{\mathbf{M}}_{kl}^R \Big/ \tau\right)$. Unlike NG-RANSAC [38], gradients are propagated through the scoring rather than the sampling and are thus much smoother. Because the sampling and scoring mirror similar featuremetric errors, negative samples become harder as the learning proceeds.

Inference-time alignment: SNAP can estimate the unknown 3-DoF relative pose between any two neural maps. We estimate each map in the sensor coordinate frame, establish tentative correspondences by matching their cells, sample pose hypotheses, and select the pose with the highest score. This includes single-image positioning, where the query map $\mathbf{M}^Q$ covers the camera frustum. The vertical pooling requires that the gravity direction is known, which is a reasonable assumption for applications like Augmented Reality (AR) and robotics [187, 264, 368]. Our framework also applies more generally to aligning any pair of inputs, including sequence-to-sequence and aerial-to-ground registration, which is required in the first place to pose mapping data in a common reference frame.

6.4. Related work

Visual positioning is most commonly tackled with geometric approaches [138, 261, 271] that rely on point correspondences across images and sparse 3D point clouds built with SfM [4, 278]. They then estimate the 6-DoF query pose with a robust solver [25, 54, 67, 68, 70, 95] from correspondences with the reference model or images. Such correspondences are most often estimated by sparse local features [11, 183]. This process is complex and end-to-end back-propagation is impractical [32]. Past works have thus focused on learning specific components like feature extraction [81, 85, 88, 185, 203, 245, 317, 330, 348, 364], matching [141, 177, 263, 303, 349, 365, 371, 373], and pose [268, 339] or point cloud refinement [176]. Coarse GNSS location and gravity direction are commonly assumed to be known [187, 306, 368]. In AR and robotics, the height of the camera can be estimated as the distance to the ground in a local SLAM reconstruction [264]. These assumptions reduce the problem to 3-DoF estimation and make it more amenable to end-to-end learning. MapNet [124] also learns end-to-end 3-DoF visual mapping and localization but requires sequences of depth inputs. Recent works leverage overhead instead of ground-level images [94, 286, 288, 356]. They easily scale to large scenes but only in open-sky areas. Their accuracy is also limited by the low resolution of aerial imagery. Our work combines the strengths of both ground-level and overhead imagery by learning end-to-end how to best fuse them for 3-DoF positioning. Our differentiable pose estimation, based on RANSAC, is more efficient [94, 124, 264], robust [288], and stable [32, 38] than previous approaches.

Semantic representations can largely benefit loop closure [280] and pose estimation [319]. OrienterNet (Chapter 5) learns 3-DoF positioning end-to-end from public 2D semantic maps that are more compact yet detailed enough for localization. Its accuracy is however limited because these maps have low spatial accuracy and are infrequently updated. It is also also restricted to few, explicit semantic classes that are often not discriminative. Differently, [161] learns finer-grained semantic classes for temporal and viewpoint consistency. Our work instead learns *implicit* semantics from posed imagery by combining end-to-end self-supervised learning with large amounts of data. This boosts the positioning accuracy and is an effective pre-training for semantic tasks.

Neural scene representation is an active topic of research. MLPs [200, 310] and tokens [258] are compact but lack geometric inductive bias. 3D voxel grids are more expressive and thus popular for reconstruction [33, 208, 229, 304, 380], rendering [207, 309], and semantic perception [31, 46, 61, 331] but are expensive to store and thus often restricted to small scenes. 2D grids, or Bird's-Eye View (BEV)s, are more compact and thus scale to larger outdoor scenes by compressing the information along the vertical axis. Neural BEVs can be learned from images for supervised semantic tasks [115, 173, 231, 249, 257], 3D reconstruction [229], self-supervised view synthesis [284], and 3-DoF positioning [94, 124, 264]. These approaches assume planar scenes or rely on monocular priors only, even if multiple views are available. Instead, we combine these priors with multi-view fusion [276, 347, 362] to leverage information from image sequences and better resolve objects in large scenes.

Self-supervised learning leverages unlabeled datasets to learn representations useful for down-stream tasks. Many works focus on image- or pixel-level contrastive learning for semantic tasks [50, 118, 119, 217, 232]. View synthesis from few images typically learns lower-level representations [200, 284]. Some works [161, 300] learn features for image matching across appearance changes. CoCoNets [158] learns representations for 3D scenes but requires perfect, synthetic depth maps. We learn high-level contrastive scene representations from posed images and show that it translates to semantic mapping.

6.5. Experiments

Data: StreetView images are captured by rigs of 6 rolling-shutter cameras mounted on cars or on backpacks worn by pedestrians [62], which results in a wide diversity of viewpoints in street-level scenes. Multi-view 'frames' are captured synchronously every ~5m. Sequences are captured between 2017 and 2022. We build mapping segments *only from car sequences* by partitioning each sequence into groups of 36 images that face either the left or right side of the road. We define each map grid as a 64×16 m tile aligned with the segment mid-frame, in which we render an aerial orthophoto with 20 cm ground sample distance. Query images are sampled

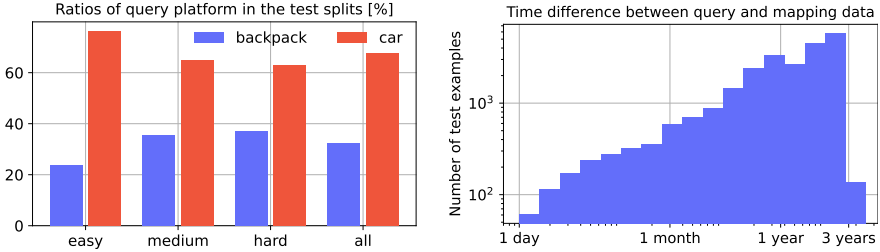


Figure 6.4.: ***Large viewpoints and temporal differences.*** *Left: The easy split contains more car-mounted queries, and the hard split contains more backpack queries. This make sense, as examples captured from backpacks are typically captured from sidewalks instead of the road, which results in difficult localization scenarios across opposite views. Right: The time difference between query and mapping images spans from a few days to a few years. This yields challenging localization scenarios with large appearance and even structural changes, e.g., due to construction work.*

from different sequences, captured from cars or backpacks, based on their frustum overlap, and are often taken years apart (Fig. 6.4).

We train with 2.5M segments and ~50M queries from 11 cities across the world: Barcelona, London, Paris (Europe), New York, San Francisco (North America), Rio de Janeiro (South America), Manila, Singapore, Taipei, Tokyo (Asia), and Sydney (Oceania), reserving some areas in each city for validation. We test on 6 different cities (Amsterdam, Melbourne, Mexico City, Osaka, São Paulo, and Seattle), with 4k queries per city. This covers 5 continents, while academic localization benchmarks focus on tourism landmarks [143] or single cities in Europe or the US [265,273,372].

Training and implementation: In the ground-level encoder, $\Phi_{\mathbf{I}}$ is a U-Net [250] with a BiT ResNet backbone [155], pre-trained as in [369], and an FPN decoder [174], initialized randomly. We consider two models with different backbones: a 'large' R152×2 (353M parameters) and a 'small' R50×1 (84M parameters). Φ_{OV} is a similarly-defined R50×1+FPN. In multi-view fusion (Sec. 6.2.2) we use $D{=}32$ depth planes and $K{=}60$ height planes $\{z_k\}$ uniformly distributed within 12 m. Neural maps $\mathbf{M}$ and matching maps $\bar{\mathbf{M}}$ have dimensions 128 and 32, respectively, and are defined over 64×16 m grids with 20 cm ground sample distance. Query BEVs have a maximum depth of 16 m. At training time, neural maps are built from one

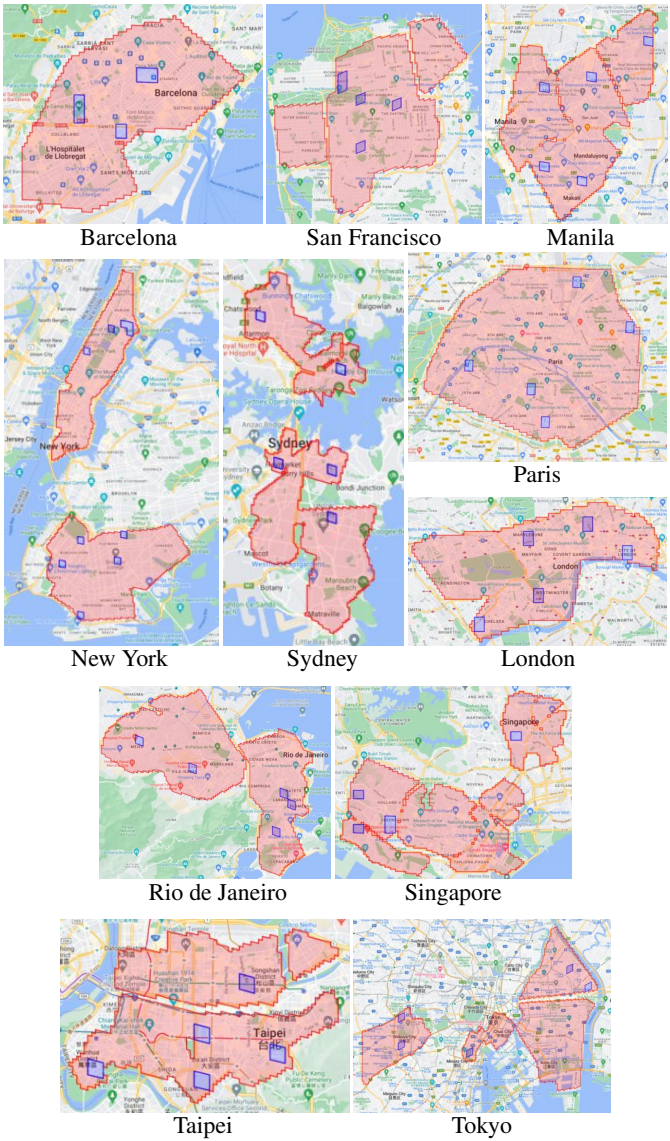


Figure 6.5.: ***Spatial distribution of the training data.*** *In each city, training examples are sampled from the* ***red*** *areas while validation examples are sampled from the* ***blue*** *areas.*

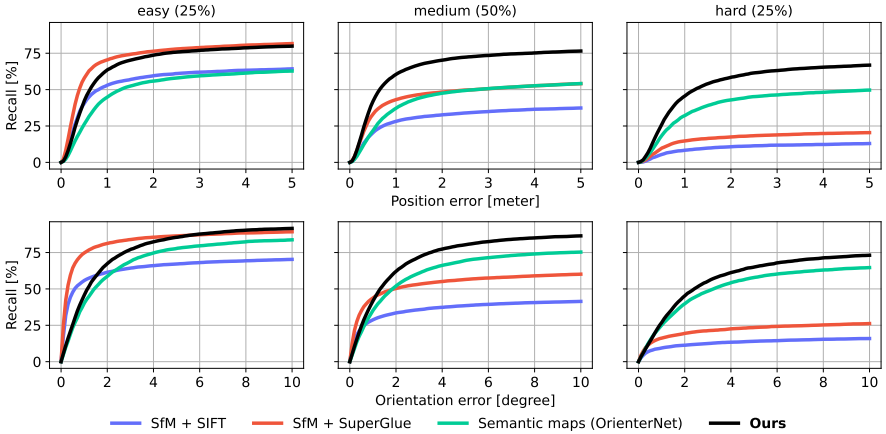


Figure 6.6.: ***Single-image positioning with different maps.*** *Localizing with our neural maps yields a higher recall than established approaches based on feature matching (SfM + X), especially for hard queries with low visual overlap. Neural maps are also more suitable for positioning than semantic maps because they encode richer and thus more discriminative information.*

aerial tile and one SV segment, with each of the two randomly dropped, similarly to dropout [301]. We use a subset of $N{=}20$ views, some of them at a $\pm 60°$ angle, which we empirically found provides a good coverage/memory trade-off.

6.5.1. Visual positioning

Setup: We build a map for each segment using all 36 views and evaluate the 3-DoF query pose in terms of position and orientation errors. While many academic benchmarks use much larger mapping areas, we argue that GNSS and motion priors often make this unnecessary for practical applications [187]. We slice the results by difficulty in terms of query-scene overlap based on the distance between the query and its closest map view, in position Δt, and orientation $\Delta\theta$. We split in the data into 3 groups: 'easy' ($\Delta t{<}10\,\text{m}$ and $\Delta\theta{<}45°$, ~25% of the data), 'hard' ($\Delta t{>}10\,\text{m}$ and $\Delta\theta{>}60°$, ~25%), and 'medium' (the remaining ~50%).

Algorithm		Inputs	Easy (25%)	Med. (50%)	Hard (25%)	All (100%)
SfM	+ SIFT [183]	StreetView	47.0 / 54.4	24.9 / 29.9	7.6 / 9.7	27.1 / 32.1
	+ SuperGlue [263]	StreetView	**63.0 / 71.1**	38.0 / 44.4	13.1 / 16.1	39.2 / 45.2
OrienterNet [264]		semantic	35.6 / 47.2	29.3 / 39.5	24.8 / 34.8	30.0 / 40.6
SNAP-large ResNet-152×2		multi-modal	48.9 / 62.3	**46.9/ 59.5**	**34.5 / 47.6**	**44.4 / 57.4**
		StreetView	45.8 / 58.4	43.9 / 56.0	29.5 / 41.7	41.0 / 53.2
		aerial	27.4 / 40.6	25.3 / 37.5	20.8 / 32.3	24.8 / 37.1
SNAP-small ResNet-50		multi-modal	45.2 / 59.0	41.9 / 54.8	29.6 / 42.0	39.9 / 52.9
		StreetView	42.2 / 54.9	38.1 / 50.1	24.5 / 36.4	36.0 / 48.2
		aerial	23.9 / 35.6	21.9 / 32.8	17.9 / 27.5	21.5 / 32.3

Table 6.1.: ***Single-image positioning.*** *We report the area under the curve (AUC) of the position and orientation errors up to thresholds (2.5 m/5°) and (5 m/10°). Our large and small multi-modal models are more accurate than classical SfM + X approaches for medium and hard queries, which matter most in practical applications. Fusing both StreetView and aerial imagery is more accurate than using only one of them.*

Baselines: We compare our approach to hloc [261], a state-of-the-art [265, 273] structure-based 6-DoF localization system based on COLMAP [278], a popular SfM framework, with correspondences estimated by either RootSIFT [11, 183] or SuperPoint+SuperGlue [81, 263], a learned feature and matcher. Note that these approaches can only leverage ground-level imagery. We match the query to all map images, without using hloc's retrieval component, and estimate the query's pose using RANSAC and a P2P solver with gravity constraint [307]. We evaluate the 6-DoF pose projected to 3-DoF. We also evaluate OrienterNet [264], which matches a query BEV with a semantic map. We re-implement and train it on overhead semantic rasters derived from SfM points with semantic labels obtained by fusing 2D image segmentations. Note that OrienterNet was originally trained on OpenStreetMap [218], which has limited coverage of small objects. While our rasters are noisy, they provide a consistent, global coverage of fine-grained classes like tree, streetlight, poles, etc. We also evaluate versions of SNAP trained with only either ground-level or aerial imagery – the latter is an extreme case of cross-view localization, similar to [94].

Results: Fig. 6.6 and Tab. 6.1 show that SNAP outperforms the state of the art, COLMAP with SuperPoint+SuperGlue, by a large margin: 25% relative. Structure-

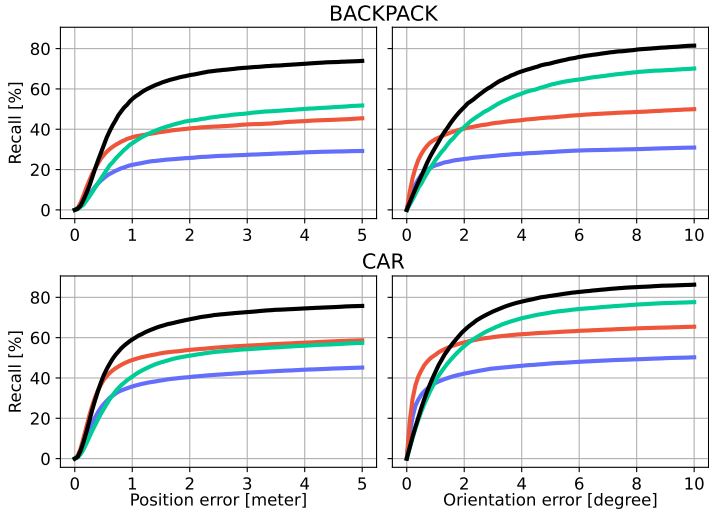


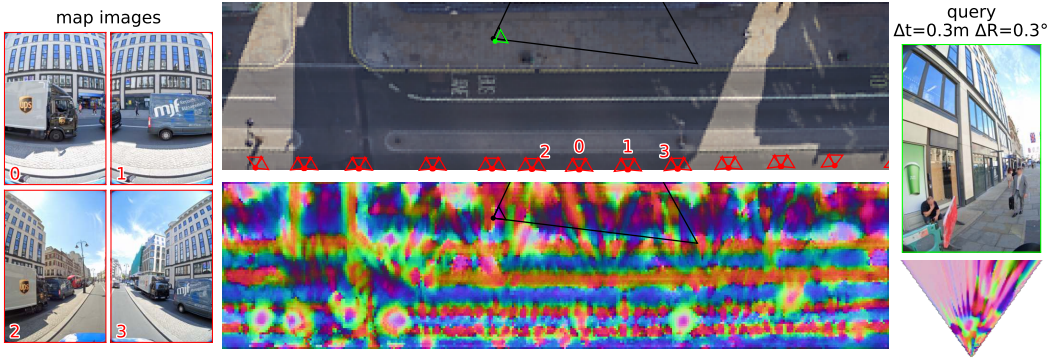
Figure 6.7.: ***Single-image positioning per platform.*** *We plot the position and orientation recall for queries taken by cameras mounted on either backpacks or cars.*

based approaches are more accurate for easy queries but significantly worse for hard ones. Using ground-level imagery is crucial in most localization scenarios and performs ~46% relative better than using aerial imagery, whereas our multi-modal model performs ~8% relative better than the StreetView-only variant.

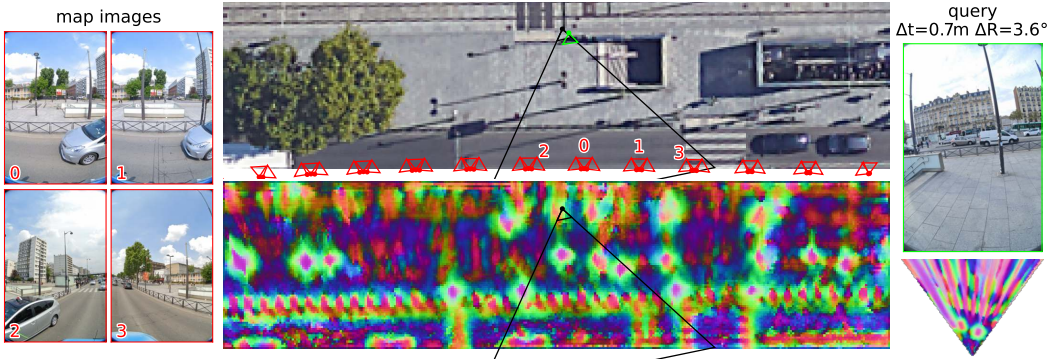
Fig. 6.7 shows the recall for queries taken by cameras mounted on either backpacks or cars. Backpack queries have a lower recall because they are typically taken from a sidewalk and thus have larger viewpoint differences (Fig. 6.4). We show qualitative examples of query successfully localized in Figs. 6.8 to 6.13.

Our framework is also efficient, as mapping takes 223 ms per segment and 6 ms per aerial tile, estimating a query BEV takes 14 ms and localizing it takes 86 ms, on an A100 GPU. In comparison, matching with SuperGlue takes 100ms per pair for 36 pairs per query, and is thus 36 times slower. Each tile of our matching maps has size 1.6 MB in fp16, while storing SuperPoint descriptors requires 5.3 MB on average.

Sequence to sequence alignment We have so far focused on the alignment of single-image queries to maps built from multiple views, but SNAP can arbitrarily

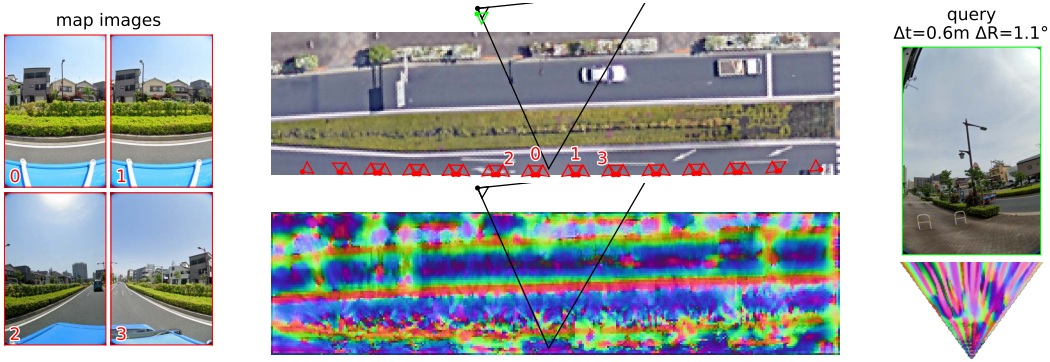


(1) Repeated structure and large viewpoint differences between query and map images (medium).

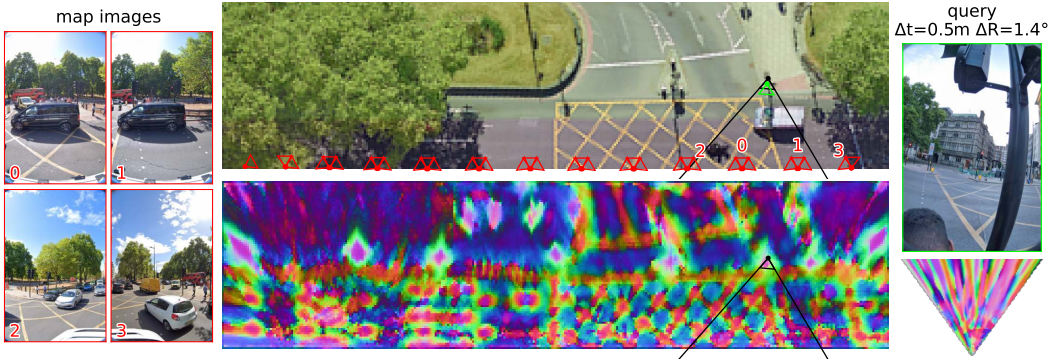


(2) Opposite views, localized using the ground and poles, observable in the neural map (hard).

Figure 6.8.: ***Single-image localization (1/6).*** *We show the 3-DoF poses of map images **(red)**, the GT query pose **(black)**, and the pose estimated by SNAP **(green)** with its error (Δt,ΔR). The error is low even for extreme opposite views. We visualize neural maps by projecting them to RGB using PCA. We can clearly recognize objects like trees, poles, curbs or road markings.*

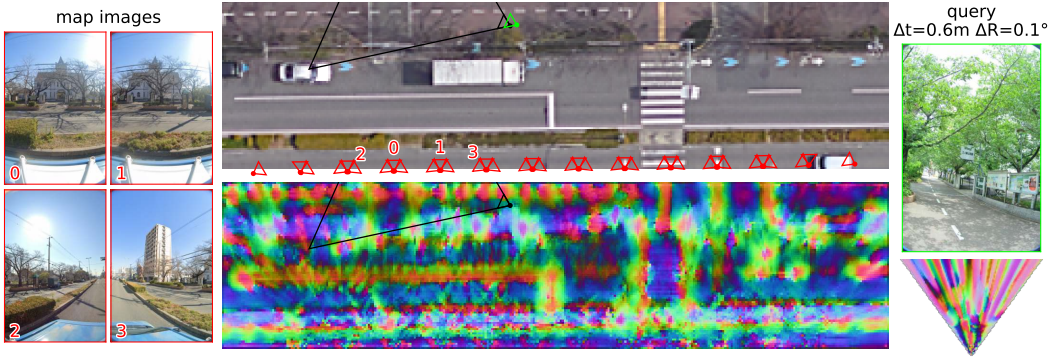


(1) Opposite views on a road scene with linear structure, localized using a pole and shrubbery (hard).

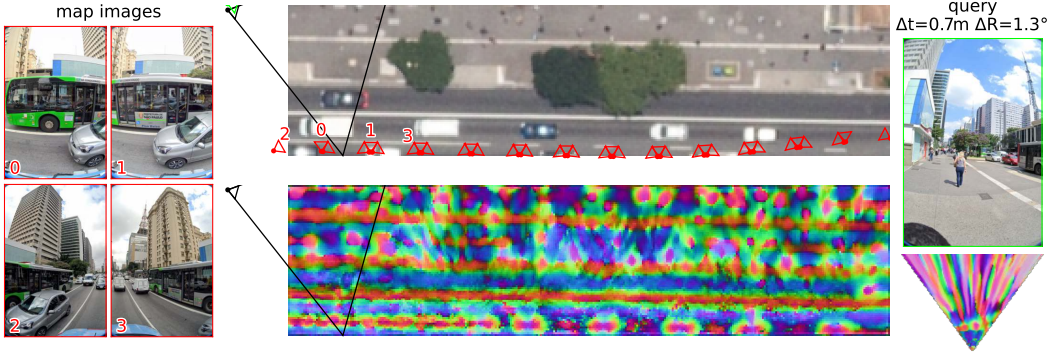


(2) Opposite views, localized using a close-up of a pole and the markings on the ground (hard).

Figure 6.9.: ***Single-image localization (2/6).*** *We show successful examples. See legend in Fig. 6.8*

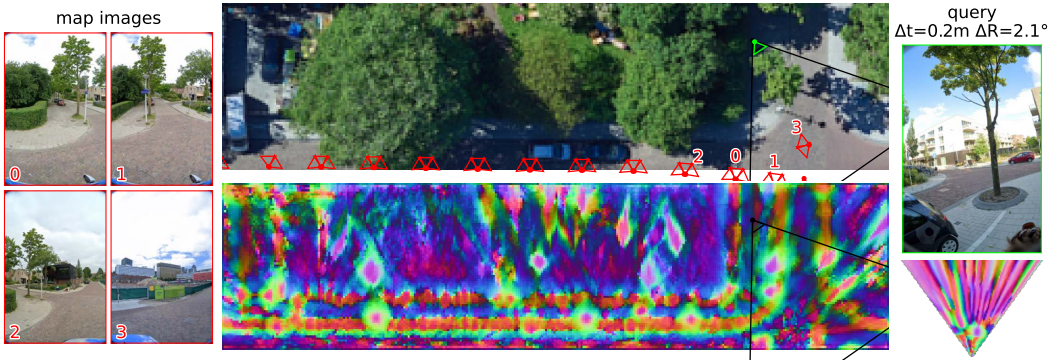


(1) The query is localized within 0.6 m from a different road (medium).

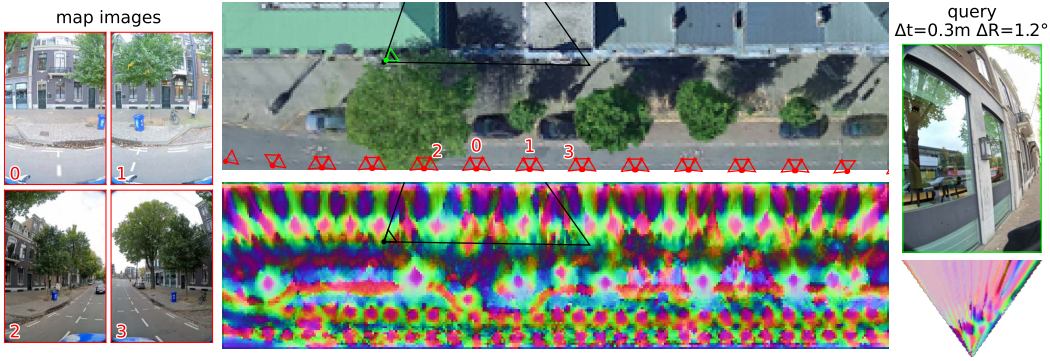


(2) The query is localized within 0.7 m despite significant occlusions in the reference views (medium).

Figure 6.10.: ***Single-image localization (3/6).*** *We show successful examples. See legend in Fig. 6.8*

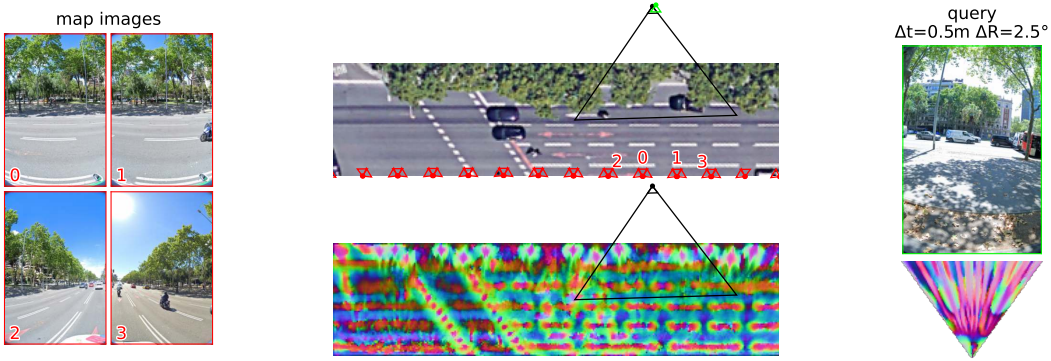


(1) The tree is clearly visible in both the query and the reference neural maps (hard).

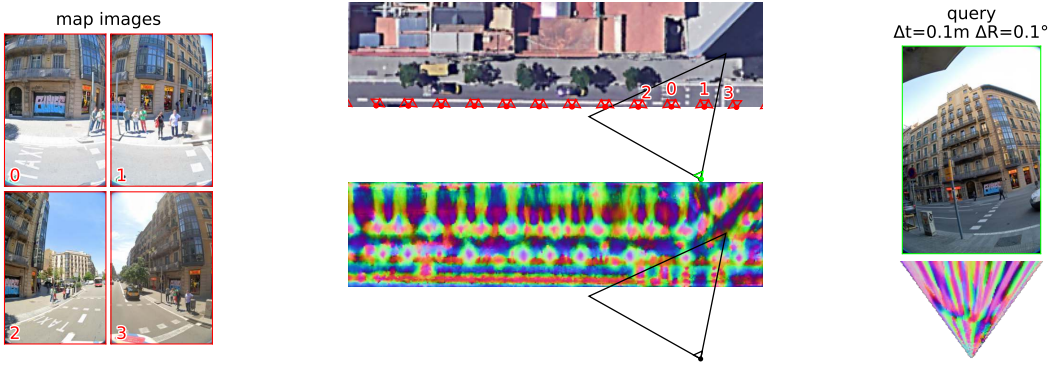


(2) The query is correctly localized despite drastic perspective changes and reflections in the window (medium).

Figure 6.11.: ***Single-image localization (4/6).*** *We show successful examples. See legend in Fig. 6.8*

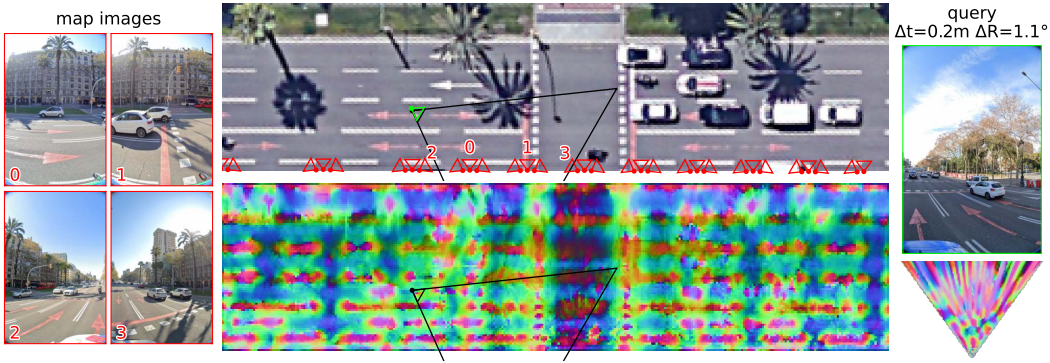


(1) The query is localized across opposite views (hard).

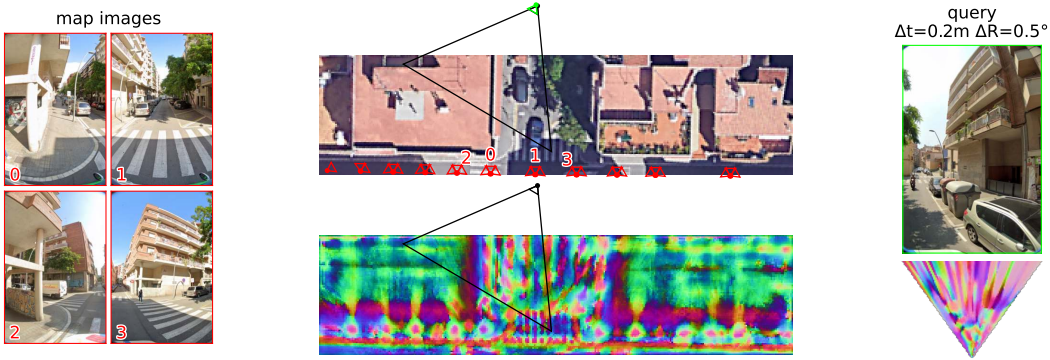


(2) The query is on the other side of the street, far away from the reference views (medium).

Figure 6.12.: ***Single-image localization (5/6).*** *We show successful examples. See legend in Fig. 6.8*



(1) Road markings are the only elements in common across opposite views (medium).



(2) The query is correctly localized across a different street and despite drastic perspective changes (hard).

Figure 6.13.: ***Single-image localization (6/6).*** *We show successful examples. See legend in Fig. 6.8*

align any pair of neural maps. We show two examples of sequence to sequence alignment in Figs. 6.14 and 6.15. These maps are built from five views. For ease of understanding, we lift our neural maps into 3D using LiDAR, color-coding each LiDAR point with the neural map values of the cell it belongs to – note that this is stricly for visualization purposes, and our algorithm does not rely on LiDAR.

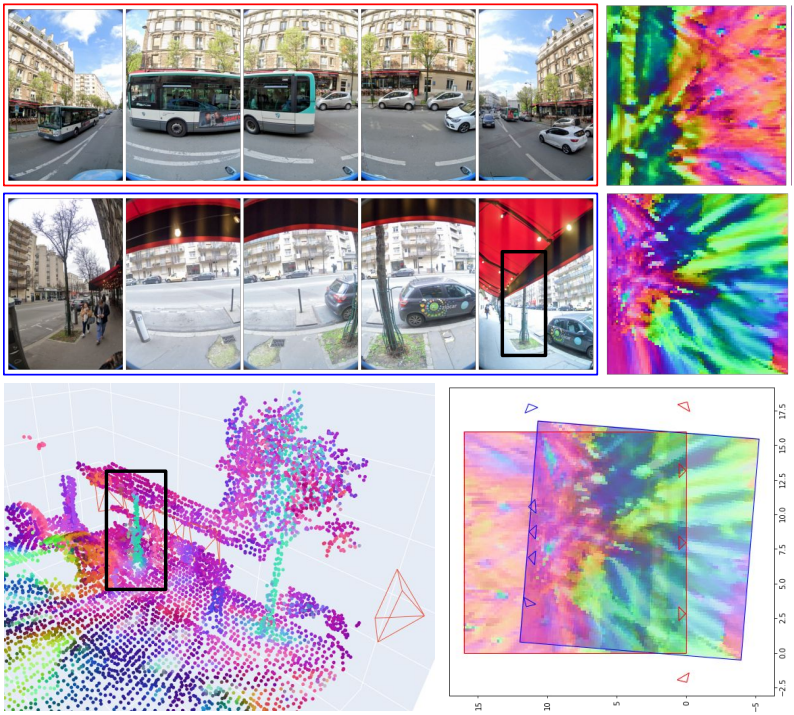
Large-scale mapping: We can easily build large tiles by 'stitching' smaller neural maps together via cell-wise max-pooling, similarly to how we fuse aerial and ground-level neural maps. Starting from multiple sequences posed in a common reference frame, Fig. 6.16 shows how neural maps are inferred for each of them and finally combined together.

6.5.2. Design decisions

In this section, we explain our design decisions and support them with an ablation study.

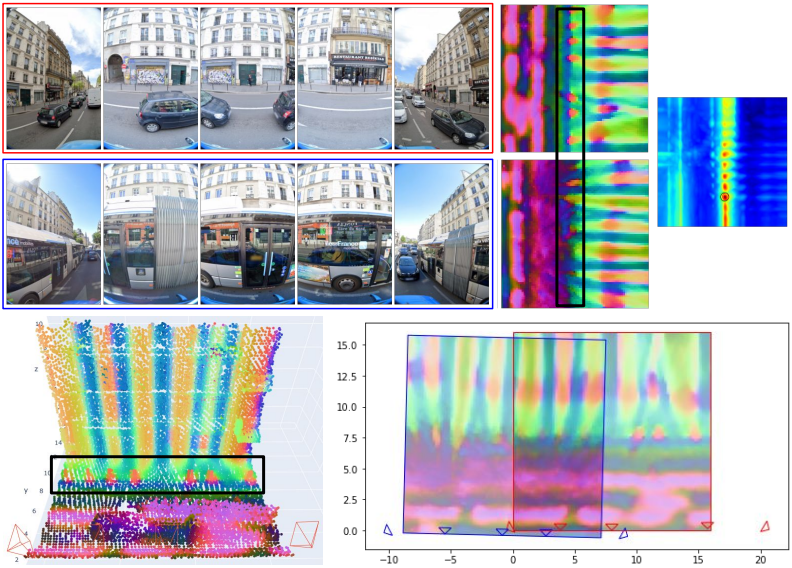
Constraints: Learning features that are discriminative requires sufficiently challenging negative pose samples. These arise from viewpoints that are visually similar to the ground truth viewpoint, for example due to repeated patterns, lack of distinctive features, or occlusions. In order to find sufficiently difficult negative samples in a training example, the map should be as large as possible. The size of the map and the number of mapping images are however limited by the amount of GPU memory available. We therefore found it critical to find the right balance between the size of the map, its spatial resolution, the number of views, and the batch size, as also reported in [115].

We found that a simple and lightweight model that saves memory is beneficial over a complex model that requires reducing the size of the map, the number of views, or the batch size. We thus favor explicit constraints by geometry (camera projection, 3D occupancy) rather than flexible but heavy mechanisms like attention or 3D convolutions. The effectiveness of our design shows that such complexity is not required. This enables the training of high-capacity models with maps of 64×16 m, at a resolution of 20 cm (~25k cells per map), from 20 images, and within 20 GB of GPU memory per example. Despite extensive memory optimizations, including the



Alignment error: Δt=0.5m, ΔR=0.33°

Figure 6.14.: ***Sequence-to-sequence alignment (1/2).*** *We align a car sequence (red) to a backpack sequence (blue) across opposite views. Notice how the neural maps (top right) are clearly similar despite the large viewpoint difference, and can be easily aligned (bottom right). In the bottom left we lift the neural map for the first sequence into 3D by coloring a LiDAR point cloud with the RGB value of the grid cell it belongs to – note that this is strictly for visualization purposes as SNAP* does not *rely on LiDAR. Notice how the semantics learned by our model correspond to real scene features, such as the pole (in cyan, enclosed in a black box).*



Alignment error: Δt=0.3m, ΔR=0.05°

Figure 6.15.: ***Sequence-to-sequence alignment (2/2).*** *We align two car scenes, the second of which is heavily occluded due to a passing bus. Notice how the neural maps (top right) are visually similar and can be easily aligned. The neural map for the first sequence clearly shows a row of poles, which are occluded in the second sequence (black box). Our method is robust and can align the sequences using road markings and building boundaries. On the top right we plot an alignment heatmap by running exhaustive aligment in 3-DoF between both neural maps (we max-pool over rotations, for ease of understanding): notice the clear maximum (circled), with smaller maxima along the road. In the bottom left we lift the neural map for the first sequence into 3D by coloring a LiDAR point cloud with the RGB value of the grid cell it belongs to – note that this is strictly for visualization purposes as SNAP* does not *rely on LiDAR.*

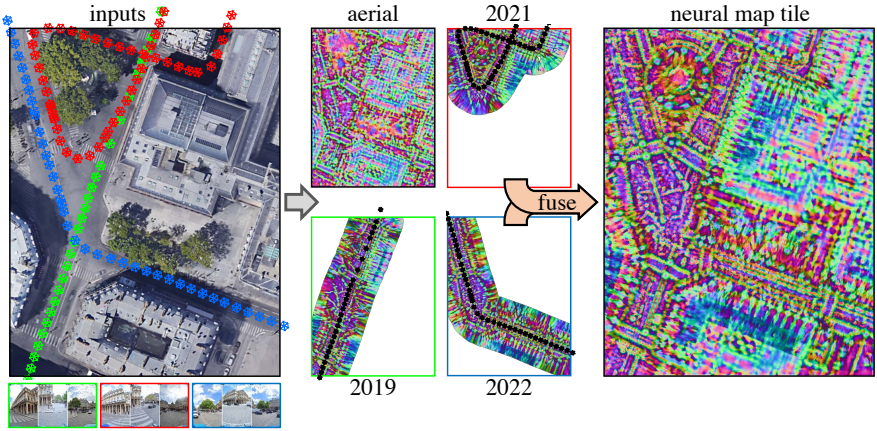


Figure 6.16.: ***Building neural tiles.*** *We combine an aerial view and car sequences captured over multiple years into a single neural map that spans a large area. An arbitrary number of inputs can be combined in such way.*

use of gradient checkpointing and mixed-precision training, using larger maps for training remains challenging.

Ground-level encoder: We consider four alternative designs for the ground-level encoder Φ_{SV}.

(a) **Fixed-height plane.** Instead of lifting the image information to 3D and pooling it vertically, the geometry of the scene can be approximated from a ground plane [1]. Driving applications commonly assume that this plane is horizontal and at a fixed height below the camera [286]. We train a variant that aggregates the multi-view information on a plane 2.5 m below the camera. This approach can hardly resolve the spatial location of overhanging structures, like street lights. It furthermore introduces distortions in the BEV if the image is not gravity aligned, which is the case for many capture platforms (such as StreetView backpacks or consumer phones), or the ground is not planar, as in many real environments.

(b) **No monocular occupancy.** Multi-view stereo [347, 362] typically does not explicitly leverage monocular geometry priors. We thus train a variant that

omits the weighting by occupancy scores. Image features are thus painted identically along all depth planes for each ray and their mean and variance across all views is fed to the fusion MLP. This model can only leverage the bearing angles of point features (like poles) to disambiguate a pose but cannot resolve the location of line features.

(c) **No multi-view variance.** SimpleBEV [115] simply averages feature volumes obtained by painting image features along each ray. This corresponds to removing both variance and monocular priors from SNAP, making it even harder to resolve surfaces.

(d) **Ray conditioning.** Close to our approach, Sharma et al. [284] augment multi-view fusion with monocular cues, but do so by conditioning each ray feature by its 3D location, using an MLP that encodes the ray direction and camera-space coordinate. This requires evaluating the MLP for each observation of each point. Our approach, based on an occupancy volume, requires only performing a tri-linear interpolation for each observation, which is significantly cheaper. We train a variant based on this MLP conditioning, replacing the weighting by occupancy score. It increases the memory requirement by 4, since we use 4 observing views per point ($|\mathcal{N}_k|=4$). We thus need to halve the batch size and the number of height planes.

We train all variants, including our model, for an identical number of steps. To save compute resources, we train for fewer steps than in the main experiment (200k vs 400k) and we use the smaller ResNet-50 architecture with only ground-level inputs. Tab. 6.2 shows that each of these variants yields a lower positioning accuracy than our model.

Vertical pooling: In our design, the vertical pooling is performed with max pooling. Tab. 6.3-top shows that average pooling is significantly less effective. We hypothesize that averaging makes it harder to ignore features of points located in empty space. We found that pooling with an attention mechanism [284] performs similarly as max pooling despite the increase in computation. Harley et al. [115] flatten the vertical elements with a space-to-depth (or pixel-shuffling) operation, followed by an MLP. This makes the model sensitive to a translation along the vertical axis, which rarely occurs in small driving datasets based on a few cars with identical specifications, but matters for heterogeneous data captured by backpacks

Variant	Easy (25%)	Med. (50%)	Hard (25%)	All (100%)
SNAP (StreetView-only)	**39.8 / 51.8**	**36.5 / 47.2**	**22.3 / 32.3**	**34.0 / 44.9**
Fixed-height plane (a)	34.0 / 45.4	29.6 / 39.8	18.6 / 28.0	28.2 / 38.6
No monocular occupancy (b)	29.8 / 41.7	26.1 / 37.3	16.1 / 25.7	24.7 / 35.8
No multi-view variance (c)	28.8 / 39.1	23.5 / 32.6	14.0 / 21.7	22.7 / 31.8
Ray conditioning (d)	11.9 / 22.5	8.7 / 17.4	4.5 / 10.1	8.6 / 17.1

Table 6.2.: ***Ablation study of the ground-level encoder.*** *We report the single-image positioning AUC up to thresholds (2.5 m/5°) and (5 m/10°). Variant (a) aggregates the information on an horizontal plane at a fixed height below the camera [1, 286]. This is insufficient for overhanging objects, when ground footprints are occluded, or when the scene is not planar. Variant (b) performs multi-view fusion without monocular priors [347, 362]. This makes it impossible to resolve the depth of objects in the single-image query. Variant (c) further drops the variance term and simply averages feature volumes [115]. This makes it harder to resolve surfaces. Variant (d) replaces the occupancy volume by conditioning each observation feature on the ray and distance using an MLP [284]. This is much more expensive and thus constrains both batch size and scene size, which in turn lowers performance.*

Component	Operator	Easy (25%)	Med. (50%)	Hard (25%)	All (100%)
Vertical pooling (StreetView-only)	max	39.8 / 51.8	36.5 / 47.2	22.3 / 32.3	34.0 / 44.9
	average	32.4 / 43.6	29.1 / 39.3	17.8 / 27.2	27.3 / 37.6
Multi-modal fusion (StreetView+aerial)	max	45.9 / 58.6	41.5 / 53.0	27.3 / 37.9	39.3 / 51.0
	average	45.5 / 58.4	40.9 / 52.6	27.8 / 38.8	39.0 / 50.9

Table 6.3.: ***Ablation study on pooling operators.*** *Top: In the ground-level encoder, pooling the features vertically with the max operator performs much better than averaging, as measured by single-image positioning AUC. Bottom: Fusing StreetView and aerial neural maps with either max or average pooling performs comparably.*

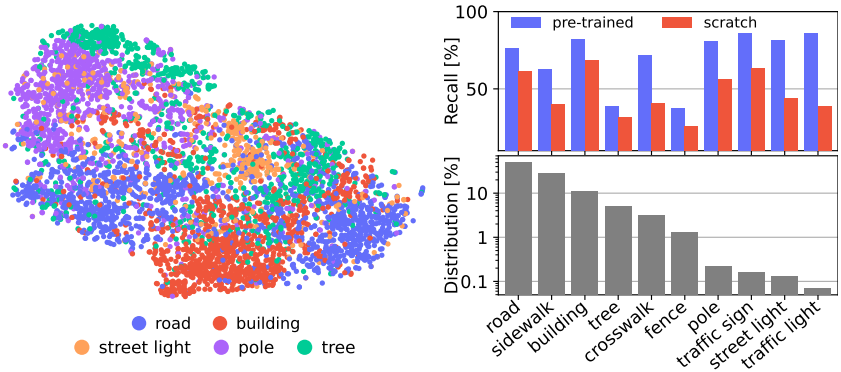


Figure 6.17.: ***2D Semantic mapping.*** *Left: t-SNE visualization of the neural map features learned by SNAP, colored by their ground-truth semantic class. SNAP discovers different categories of objects common in outdoor urban scenes, which yields clearly distinguishable clusters. Right: Given a small labeled dataset, training a tiny CNN classifier to predict such classes from pre-trained features is more effective than training the entire SNAP model from scratch, especially for small and infrequent objects.*

and cars with widely different setups. We thus found that this approach yields a lower performance than a simpler pooling. This also makes it impossible to adjust the number of height planes at inference time.

Multi-modal fusion: Tab. 6.3-bottom shows that the choice of pooling operator makes little difference when fusing neural maps inferred from StreetView and aerial inputs.

6.5.3. Semantic mapping

We show that SNAP's neural maps are an effective pre-training for 2D semantic mapping.

Qualitative inspection: Existing approaches rely on ground truth 2D semantic rasters derived from the segmentation of LiDAR 3D point clouds. These are manually labeled, which is too expensive to generate enough data to train from scratch models that generalize across countries, sensors, seasons, and times of the day.

Existing datasets [31, 42] thus rarely span more than a few cities and overfit supervised models to the local appearance. Instead, our self-supervised pre-training learns better features from a much larger dataset of posed imagery, which is much cheaper to acquire at scale. The information bottleneck forces SNAP to learn unified representations for objects, like street crossings or lights, that look very different across countries, and would require larger amounts of labeled data. Fig. 6.17-left shows a 2D t-SNE [336] visualization of SNAP's neural maps at points sampled on a few types of objects common in street scenes, according to their ground-truth semantic label. Points of the same class are clustered together. This clearly shows that neural maps learn to distinguish these objects without any semantic supervision, even if they are geometrically similar, e.g., tree *vs* pole.

Quantitative evaluation: To evaluate the pre-training, we train a tiny CNN to predict semantic rasters from pre-trained neural maps, keeping SNAP frozen. We compare this to training the entire model from scratch (with the same backbones initialization [155, 369]). We derive 3k 64×16 m ground truth rasters from LiDAR point clouds captured by StreetView cars in 84 cities across the world. We train with 2k examples and report the recall of both approaches on 1k test examples in Fig. 6.17-right. Pre-training consistently yields better results for every class, with larger gains on more difficult/infrequent classes. While training from scratch massively overfits to such small dataset, our neural maps encode enough information to reach recalls over 70%. We show qualitative examples in Figs. 6.18 to 6.21.

Monocular priors: We visualize in Fig. 6.22 the occupancy predicted by SNAP as depth and confidence maps. SNAP learns sensible priors over the geometry of street scenes from only pose supervision.

Figure 6.18.: *Semantic mapping (1/4).* *2D segmentation prediction from pre-trained neural maps by a small CNN trained in a supervised fashion with GT semantic labels of 2k scenes. The last row shows 3D a lidar point cloud colored by the prediction segmentation (lidar is not used as input). In this example, the curb is mostly occluded by the parked cars and has very sparse ground truth labels. Notice how SNAP's neural maps encode the location of street lights that are hanging above the street.*

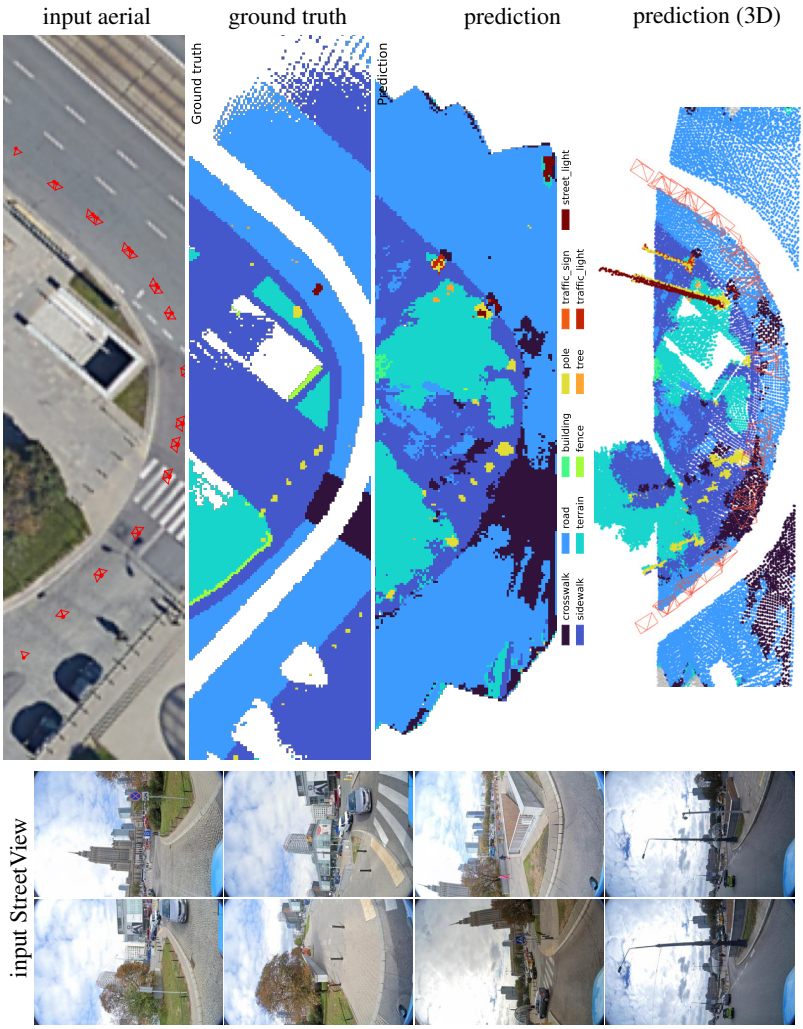


Figure 6.19.: ***Semantic mapping (2/4).*** *See legend in Fig. 6.18. In this example, because of the bike lane and confusing markings at the intersection, the model incorrectly segments the crosswalk.*

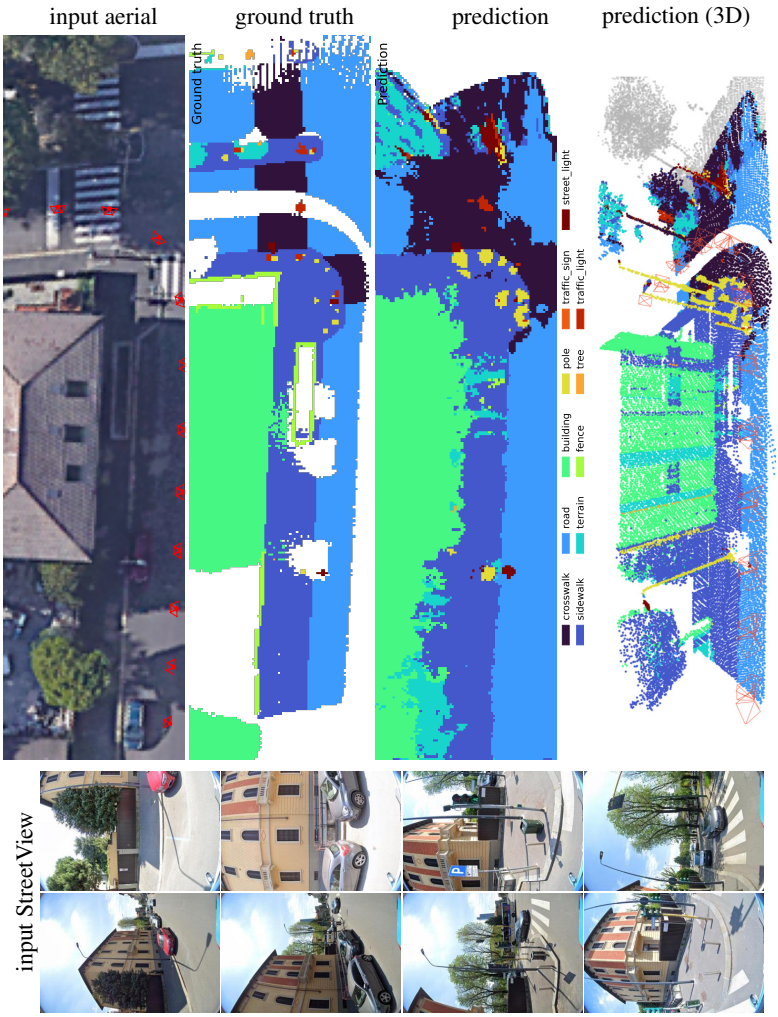


Figure 6.20.: *Semantic mapping (3/4).* *See legend in Fig. 6.18. In this example, the StreetView car turns around the corner, so the right side of the map is behind the cameras and thus very sparsely observed. Areas with good visual coverage are much better segmented.*

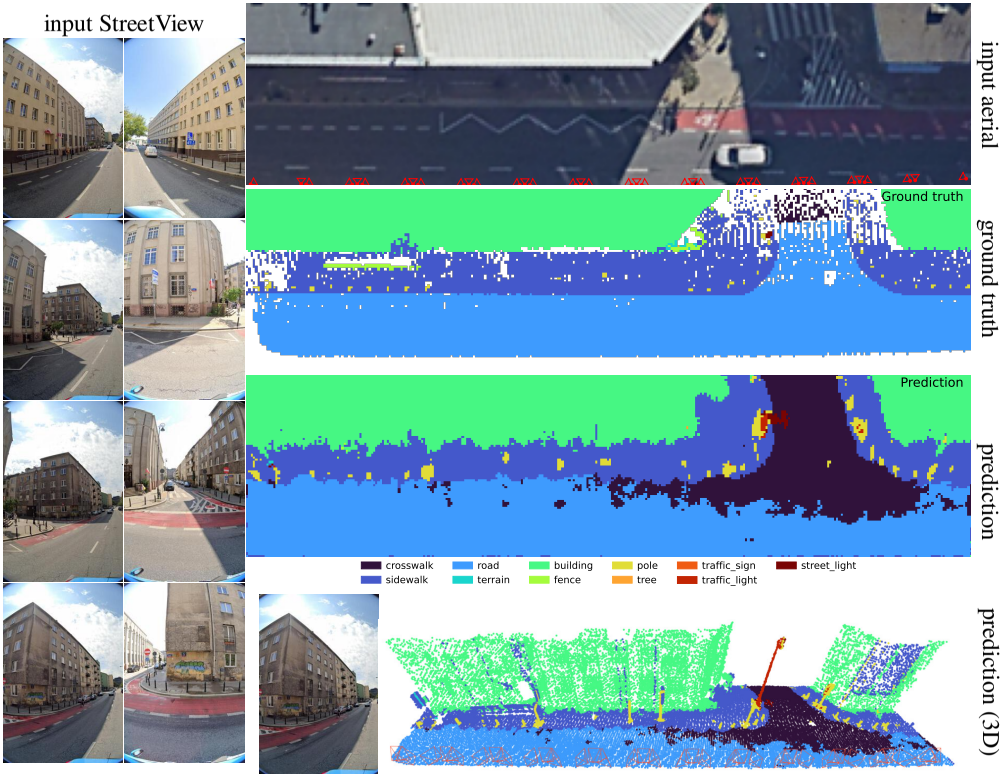


Figure 6.21.: ***Semantic mapping (4/4).*** *See legend in Fig. 6.18. In this example, the model roughly segments road, sidewalk and buildings, and correctly resolves small objects such as the poles or the overhanging traffic light. It gets confused around the sidewalk, which has multiple striped patterns that are also difficult to interpret by the human eye.*

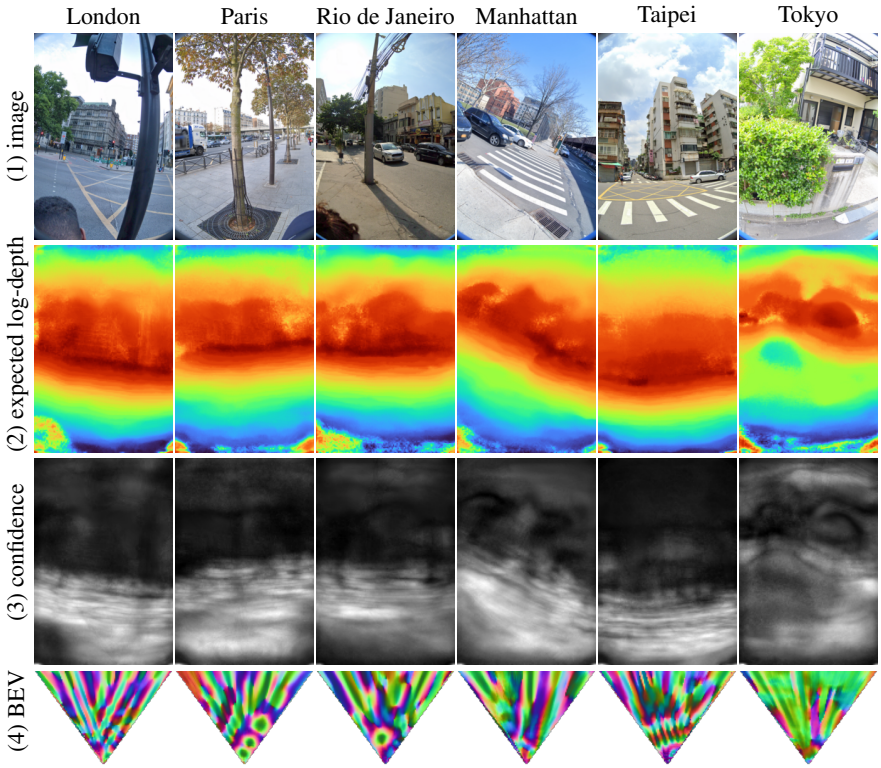
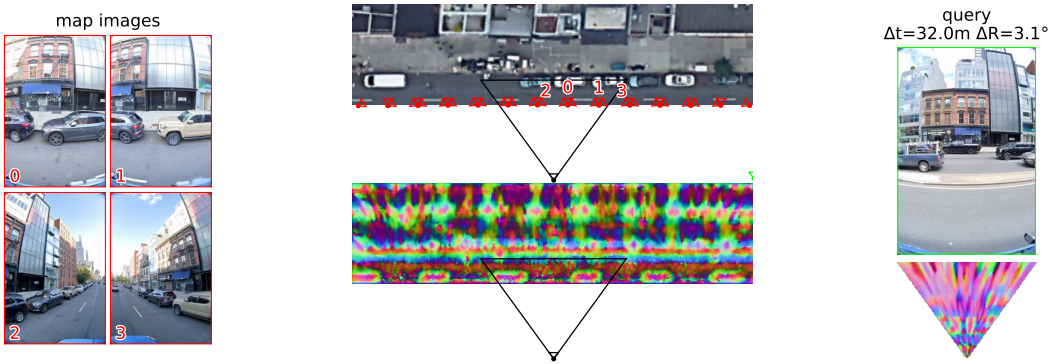
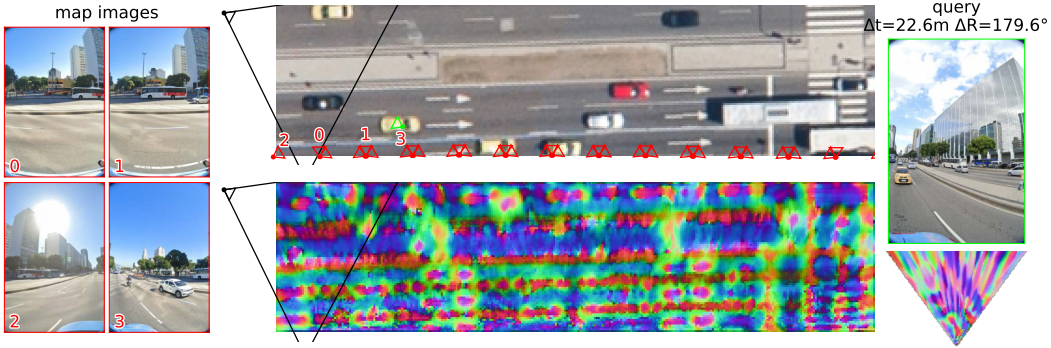


Figure 6.22.: ***Monocular depth priors learned by the ground-level encoder.*** *For each query image (1), we show: (2) the expected log-depth across all depth planes, from blue (close) to red (distant); (3) the total score along each ray* $\log \sum_{i \in \{1 \dots D\}} \exp \mathbf{S}\,[:, :, d_i]$*, which reflects how useful or confident the prediction is; (4) the resulting bird's-eye view. The predictions are sensible for areas close to the ground and for lower parts of objects and buildings. Predictions in the sky and upper facades are not reliable because these areas are never covered by the height planes* $\{z_k\}$ *of the point columns.*

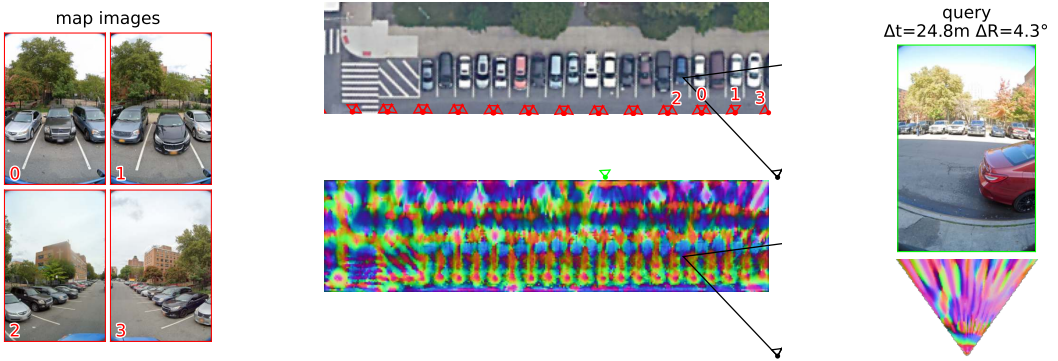


(1) The query is too far from the buildings and fails to localize using features along the road (medium).

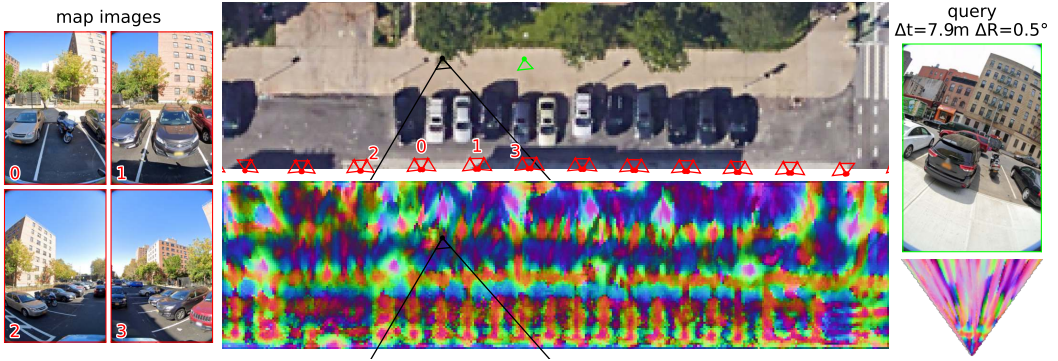


(2) SNAP match the right visual content from the wrong side, due to scene symmetries (medium).

Figure 6.23.: ***Localization failures (1/4).*** *See legend in Fig. 6.8.*

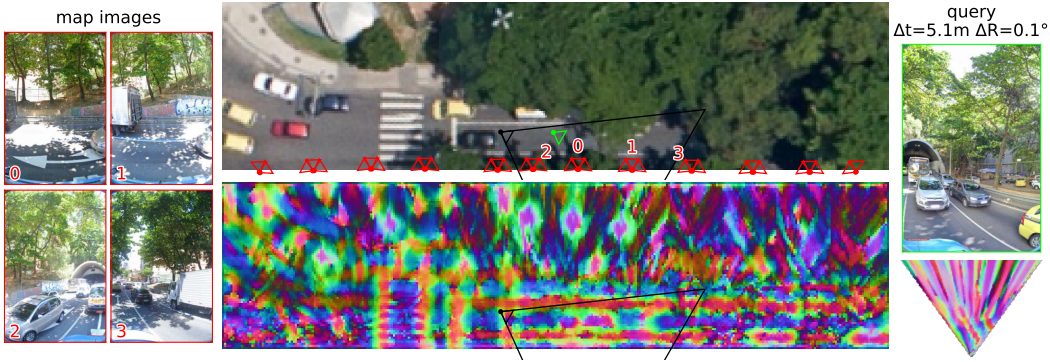


(1) Reasonable but incorrect guess along a row of parking spots with very little structure otherwise (medium).

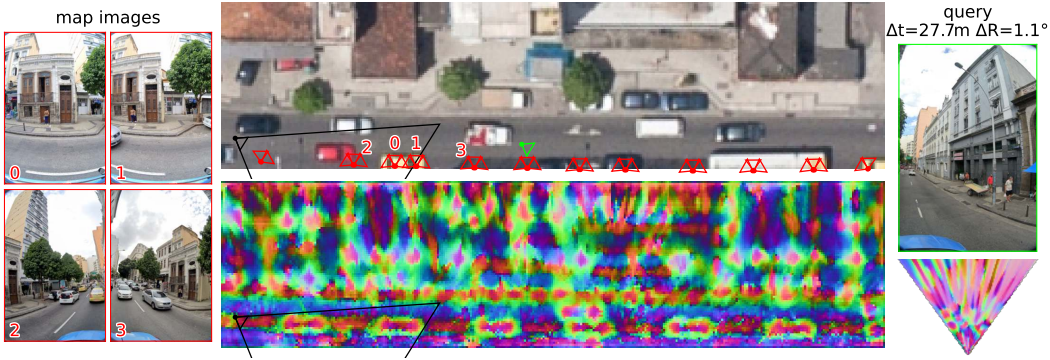


(2) Similar to (c), but across opposite views, and thus more challenging. Note that the rotation error is ¡1° (hard).

Figure 6.24.: ***Localization failures (2/4).*** *See legend in Fig. 6.8.*

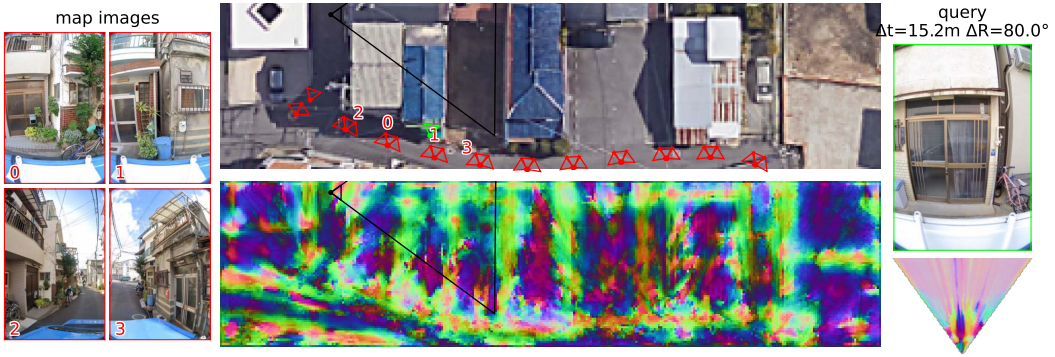


(1) A difficult case with shadows and occluders (medium).

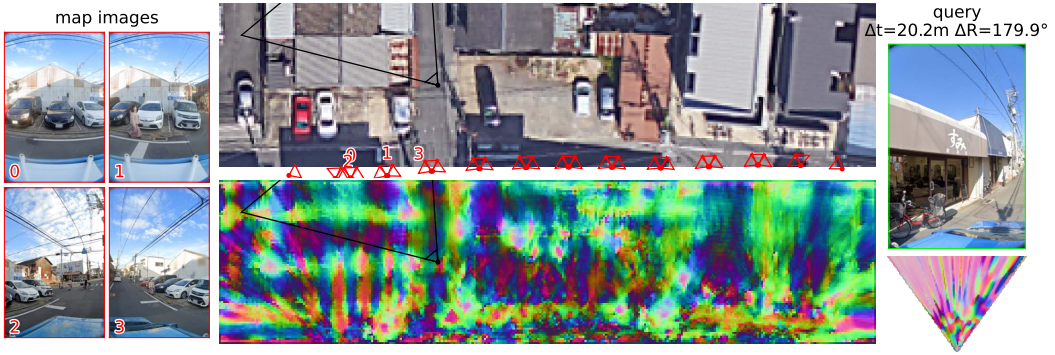


(2) Correct orientation (1°) at the wrong location along the road, across opposite query/reference views (medium).

Figure 6.25.: ***Localization failures (3/4).*** *See legend in Fig. 6.8.*



(1) [Osaka] Query and reference views are very close to the buildings, which is rarely seen during training (easy).



(2) [Osaka] Query is on a side street with little visual overlap with the reference views (easy).

Figure 6.26.: ***Localization failures (4/4).*** *See legend in Fig. 6.8.*

6.6. Summary and outlook

Summary: We present SNAP, a novel approach to build semantic, 2D neural maps from multi-modal inputs and train it by simply learning to align two neural maps in a contrastive framework. This simple objective yields a model that can localize queries beyond the reach of the state of the art in structure-based matching by discovering high-level semantics from self-supervision. Our neural maps are easily interpretable and provide an effective pre-training towards unlocking semantic understanding at scale.

Limitations and outlook: Because this work is based on data owned by Google and not publicly available, reproducing the results presented in this chapter will be difficult. Our approach is not as accurate as approaches based on 3D maps given easy queries closer to the mapping images (Fig. 6.6). We hypothesize this is partly due to operating at lower image and map resolutions. It also assumes gravity direction and a location prior, which are reasonable assumptions but restrict its use.

The semantic information learned by SNAP is limited because it is trained on relatively small maps. There is so much more information in the world that could be leveraged for a more robust localization. Examples include the type of businesses visible in the streets and the appearance of their storefronts, the type of trees, the shape and height of the buildings, or the name of public transport stations. We believe that increasing the size of these maps will make the learning problem significantly more challenging and push the model to learn some of these stronger semantic cues.

One structural limitation of SNAP is that it cannot leverage visual information that is distant and thus outside of the BEV. Its area is limited by the computational resources available. There are multiple avenues for increasing it, such as introducing some sparsity or a hierarchical representation that stores most of the information at a coarser spatial resolution. This is left to future works.

CHAPTER 7

Conclusions

7.1. Summary

In the course of this thesis, we explored the challenges of combining deep learning and 3D geometry for problems related to visual localization and mapping. Starting with existing systems that rely on 3D maps, we derived new algorithms based on the multi-view alignment of features learned by DNNs. Next, we studied how to leverage more compact and interpretable 2D maps, with new algorithms based on semantic maps and on neural maps estimated from raw imagery in an end-to-end manner. We summarize the contributions of this thesis:

Chapter 2 showed that Structure-from-Motion can highly suffer from the lack of accuracy inherent to single-view keypoint detection. To remedy this, we introduced a refinement process that aligns deep features across multiple views. Compared to photometric costs typical used in previous works, we found that features learned by off-the-shelf DNNs are more robust to appearance changes and provide a wider basin of convergence. We designed a system that leverages them in an efficient and scalable manner.

Chapter 3 showed that such features can be learned end-to-end for the process of pose refinement, resulting in higher robustness to temporal changes. The resulting DNN is a new kind of localization approach that refines the pose of the query image starting from a coarse initial estimate obtained, e.g., with image retrieval. We rely on a differentiable optimization process such that the only

learnable component is the CNN that predicts the features. This results in better generalization, accuracy, interpretability, and can be applied to new scenes without retraining.

Chapter 4 focused on a common application of localization and mapping: Augmented Reality. We noted that existing academic benchmarks are not representative of the typical data captured by AR devices. These benchmarks are often based on small-scale datasets with low scene diversity, captured from stationary cameras, and lack the diversity of sensor inputs that can be found on AR devices, such as inertial, radio, or depth data. Since benchmarking is a critical aspect of research that fuels progress in Computer Vision, we introduced a new benchmark and associated dataset. Our evaluation showed the large potential of leveraging other sensor modalities like radio signals and inertial measurements or temporal sensor streams like image sequences instead of single images.

Chapter 5 jumped into the realm of 2D maps. We note that humans can easily navigate their environment using 2D semantic maps but that algorithms have so far been unable to leverage them effectively. We derived a new learning algorithm that leverages knowledge of 3D geometry, in terms of both camera calibration and gravity information, to localize an image with sub-meter accuracy within the same 2D planimetric maps that humans use. It mimics the way humans orient themselves in their environment by matching the input map with a mental map derived from visual observations. We have shown that this algorithm can successfully localize single images in environments with rich semantic features, and can otherwise effectively leverage sequences of images.

Chapter 6 extended this paradigm to also learn the generation of better maps to maximize the accuracy of visual positioning. This does not rely on semantic maps but instead on raw imagery that is already available in many parts of the world. We introduced an algorithms that fused multi-view information from both ground-level and aerial imagery into rich neural maps. These maps encode not only geometry and appearance but also high-level semantics, discovered without explicit supervision. Consequently, we found that the resulting localization algorithm resolve the location of challenging image queries beyond the reach of existing approaches based on 3D maps. Moreover, this enables an effective

pre-training for data-efficient semantic scene understanding, with the potential to unlock cost-efficient creation of more detailed maps.

Throughout this thesis, I attempted to show that more careful handling of 3D geometry, for example in terms of camera calibration or optimization, results in substantial benefits over simpler alternatives that are based on black-box DNNs. I also strove to show that leveraging geometry does not preclude end-to-end training. Combining both helps learning stronger data-driven priors that optimized for the end-task of localization and mapping, but also ensures that the algorithm can generalize to new operating conditions. This requires a careful design such that the only priors that are learned are those that cannot be easily modeled mathematically. This enhances the interpretability of such algorithms and makes it easier to detect their failure. This is a critical feature of any systems that aims to be applied to real-world data and integrated in practical applications.

7.2. Outlook

The algorithms proposed in this thesis have advanced the robustness and accuracy of visual localization and mapping with respect to the challenges mentioned in Sec. 1.2. There is however ample room for building even more powerful algorithms and tackling related tasks. We give here a non-exhaustive list of interesting research directions.

Learning stronger 3D prior: The mapping algorithm presented in Chapter 2 improves the accuracy of the SfM and the localization algorithm presented in Chapter 3 improves robustness to long-term visual and structural temporal changes. They still struggle in the most critical conditions that arise, for example, from very sparse views with low visual overlap or from extreme temporal and viewpoint changes. These algorithms indeed still have very limited learning capacity. Tackling these challenges requires additional priors about the shape and the regularities of the real world.

We now know that DNN are increasingly good at inferring observations like monocular depth or surface normals, which are useful local single-image priors. How to best integrate such priors into SfM and SLAM, ideally in an end-to-end training process,

remains an open question. Additional higher-level priors could be leveraged: priors about camera motions or distributions, about the planarity of surfaces, or about the symmetry of common objects and structures. It is unclear how such priors can be learned and leveraged as constraints within the existing optimization processes of 3D geometry.

Leveraging multi-sensor streams: As seen in Chapter 4, AR devices and robots often carry additional onboard sensors besides cameras, such as depth sensors, inertial measurement units, and radio receivers for GNSS, Bluetooth, and Wifi. These sensors provide valuable information that is complimentary to images. They are also available as long temporal streams. These opportunities are often overlooked in existing benchmarks and algorithms. We have proposed simple approaches to leverage them, which have already shown clear benefits.

There is a lot of room for further improvements. DNNs could be particularly suitable for cross-modality data association and fusion. Since not all modalities are always available, this requires flexible algorithms that can handle missing data and heterogeneous inputs. Additionally, given the wide diversity of sensors, algorithms should leverage prior sensor calibration parameters, which are often available. This requires a tight integration with 3D geometry and physical models, which is in line with the direction taken by this thesis.

Self-supervised 2D maps: Chapters 5 and 6 showed that comparing and fusing data across visual viewpoints and semantic types is much easier in 2D than with sparse 3D point clouds. This self-supervised scheme could be extended to additional sensor modalities like LiDAR, radio data, or multi-spectral imagery. This could enable learning maps with much richer information with a wider range of use cases across remote sensing, robotics, or computer graphics. More practically, this could also enable the automated update and improvement of 2D semantic maps with only raw imagery. How to effectively turn our neural maps into compact semantic vector elements remains an open research question.

We believe that the thoughtful combination of learning and 3D geometry will be a valuable asset in solving these problems.

References

[1] Ammar Abbas and Andrew Zisserman. A Geometric Approach to Obtain a Bird's Eye View From an Image. In *Proc. of the International Conf. on Computer Vision (ICCV) Workshops*, 2019.

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[3] Sameer Agarwal, Keir Mierle, and Others. Ceres Solver. `http://ceres-solver.org`.

[4] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle Adjustment in the Large. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010.

[5] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. Rolling Shutter Absolute Pose Problem With Known Vertical Direction. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric Bundle Adjustment for Vision-Based SLAM. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2016.

[7] Hatem Alismail, Brett Browning, and Simon Lucey. Robust tracking in low light and sudden illumination changes. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2016.

[8] Manuel López Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary Planet-Scale Depth Dataset. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[9] Epameinondas Antonakos, Joan Alabort-i Medina, Georgios Tzimiropoulos, and Stefanos P Zafeiriou. Feature-based lucas–kanade and active appearance models. *IEEE Trans. on Image Processing (TIP)*, 2015.

[10] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] Anil Armagan, Martin Hirzer, Peter M Roth, and Vincent Lepetit. Accurate Camera Registration in Urban Environments Using High-Level Feature Matching. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2017.

[13] Anil Armagan, Martin Hirzer, Peter M Roth, and Vincent Lepetit. Learning to align semantic segmentation and 2.5D maps for geolocalization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] Hernán Badino, D Huber, and Takeo Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium*, pages 794–799. IEEE, 2011.

[15] Hernan Badino, Daniel Huber, and Takeo Kanade. The CMU Visual Localization Data Set. `http://3dvis.ri.cmu.edu/data-sets/localization`, 2011.

[16] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[17] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM*, 2000.

[18] Simon Baker, Ralph Gross, and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision (IJCV)*, 56, 2003.

[19] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous metric learning relocalisation using neural nets. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[21] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2016.

[22] Daniel Barath, Luca Cavalli, and Marc Pollefeys. Learning to find good models in RANSAC. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[23] Daniel Barath, Jiri Matas, and Jana Noskova. MAGSAC: Marginalizing Sample Consensus. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[24] Daniel Barath, Dmytro Mishkin, Ivan Eichhardt, Ilia Shipachev, and Jiri Matas. Efficient Initial Pose-graph Generation for Global SfM. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[25] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiří Matas. MAGSAC++, a Fast, Reliable and Accurate Robust Estimator. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] Dan Barnes, Rob Weston, and Ingmar Posner. Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information. In *Proc. Conf. on Robot Learning (CoRL)*, 2020.

[27] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Keypoint Detection by Handcrafted and Learned CNN Filters. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] Ioan Andrei Barsan, Shenlong Wang, Andrei Pokrovsky, and Raquel Urtasun. Learning to Localize Using a LiDAR Intensity Map. In *Proc. Conf. on Robot Learning (CoRL)*, 2018.

[29] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 2008.

[30] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006.

[31] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[32] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced Feature Points: Optimizing Feature Detection and Description for a High-Level Task. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] Aljaž Božič, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. TransformerFusion: Monocular RGB Scene Reconstruction using Transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[34] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[35] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-Differentiable RANSAC for Camera Localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] Eric Brachmann and Carsten Rother. Learning Less is More-6D Camera Localization via 3D Surface Regression. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[37] Eric Brachmann and Carsten Rother. Expert Sample Consensus Applied to Camera Re-Localization. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[38] Eric Brachmann and Carsten Rother. Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[39] Eric Brachmann and Carsten Rother. Visual Camera Re-Localization from RGB and RGB-D Images Using DSAC. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2021.

[40] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. A General Solution To The P4P Problem for Camera With Unknown Focal Length. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[41] Wolfram Burgard, Dieter Fox, Daniel Hennig, and Timo Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 1996.

[42] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[43] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010.

[44] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the Blind Perspective-n-Point Problem End-To-End With Robust Differentiable Geometric Optimization. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[45] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid Scene Compression for Visual Localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[46] Anh-Quan Cao and Raoul de Charette. MonoScene: Monocular 3D Semantic Scene Completion. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[47] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying Deep Local and Global Features for Image Search. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[48] Song Cao and Noah Snavely. Minimal Scene Descriptions from Structure from Motion Models. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[49] Nicholas Carlevaris-Bianco, Arash K. Ushani, and Ryan M. Eustice. University of Michigan North Campus long-term vision and lidar dataset . *International Journal of Robotics Research (IJRR)*, 2015.

[50] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[51] Francesco Castaldo, Amir Zamir, Roland Angst, Francesco Palmieri, and Silvio Savarese. Semantic cross-view matching. In *Proc. of the International Conf. on Computer Vision (ICCV) Workshops*, 2015.

[52] 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 *Proc. of the International Conf. on 3D Vision (3DV)*, 2019.

[53] 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 *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[54] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. AdaLAM: Revisiting Handcrafted Outlier Detection. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[55] Tat-Jen Cham, Arridhana Ciptadi, Wei-Chian Tan, Minh-Tri Pham, and Liang-Tien Chia. Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[56] Yiu-Tong Chan, Wing-Yue Tsui, Hing-Cheung So, and Pak chung Ching. Time-of-arrival based localization under NLOS conditions. *IEEE Transactions on Vehicular Technology*, 2006.

[57] Che-Han Chang, Chun-Nan Chou, and Edward Y Chang. CLKN: Cascaded Lucas-Lanade networks for image alignment. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[58] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[59] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-End Learnable Geometric Vision by Backpropagating PnP Optimization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[60] David M. Chen, Georges Baatz, Kevin Köser, Sam S. Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, Bernd Girod, and Radek Grzeszczuk. City-scale landmark identification on mobile devices. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[61] Ricson Cheng, Ziyan Wang, and Katerina Fragkiadaki. Geometry-Aware Recurrent Neural Networks for Active Visual Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[62] Danny Cheung. Mapping stories with a new Street View Trekker. `https://blog.google/products/maps/mapping-stories-new-street-view-trekker/`, 2018.

[63] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[64] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal Correspondence Network. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[65] Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. UnsuperPoint: End-to-end unsupervised interest point detector and descriptor. *arXiv:1907.04011*, 2019.

[66] Hang Chu, Andrew Gallagher, and Tsuhan Chen. GPS Refinement and Camera Orientation Estimation from a Single Image and a 2D Map. In *Proc.*

of the Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops, 2014.

[67] Ondřej Chum and Jiri Matas. Matching with PROSAC - Progressive Sample Consensus. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[68] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally Optimized RANSAC. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2003.

[69] Ondřej Chum and Jiří Matas. Optimal Randomized RANSAC. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 30(8):1472–1482, 2008.

[70] Ondřej Chum, Tomas Werner, and Jiří Matas. Two-View Geometry Estimation Unaffected by a Dominant Plane. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[71] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. LS-Net: Learning to solve nonlinear least squares for monocular stereo. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[72] David Cohen-Steiner and Frank Da. A greedy Delaunay-based surface reconstruction algorithm. *The visual computer*, 20(1):4–16, 2004.

[73] Ciprian-Romeo Comsa, Jianghong Luo, Alexander Haimovich, and Stuart Schwartz. Wireless Localization using Time Difference of Arrival in Narrow-Band Multipath Systems. In *2007 International Symposium on Signals, Circuits and Systems*, 2007.

[74] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual Categorization with Bags of Keypoints. In *Proc. of the European Conf. on Computer Vision (ECCV) Workshops*, 2004.

[75] Jan Czarnowski, Stefan Leutenegger, and Andrew J. Davison. Semantic Texture for Robust Dense Tracking. In *Proc. of the International Conf. on Computer Vision (ICCV) Workshops*, 2017.

[76] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[77] Philip David and Sean Ho. Orientation descriptors for localization in urban environments. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[78] Andrew J Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2003.

[79] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[80] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[81] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[82] Frédéric Devernay and Olivier D Faugeras. *Computing differential properties of 3-D shapes from stereoscopic images without 3-D models*. 1994.

[83] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. CamNet: Coarse-to-fine retrieval for camera re-localization. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[84] Mihai Dusmanu, Ondrej Miksik, Johannes Lutz Schönberger, and Marc Pollefeys. Cross-Descriptor Visual Localization and Mapping. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[85] 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 *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[86] Mihai Dusmanu, Johannes L. Schönberger, and Marc Pollefeys. Multi-View Optimization of Local Feature Geometry. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[87] Mihai Dusmanu, Johannes L. Schönberger, Sudipta N. Sinha, and Marc Pollefeys. Privacy-Preserving Image Features via Adversarial Affine Subspace Embeddings. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[88] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond Cartesian Representations for Local Descriptors. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[89] Ivan Eichhardt and Daniel Barath. Optimal Multi-view Correction of Local Affine Frames. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2019.

[90] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[91] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014.

[92] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, Cheng Peng, Chris Sweeney, Cole Wilson, Dan Barnes, Daniel DeTone, David Caruso, Derek Valleroy, Dinesh Ginjupalli, Duncan Frost, Edward Miller, Elias Mueggler, Evgeniy Oleinik, Fan Zhang, Guruprasad Somasundaram, Gustavo Solaira, Harry Lanaras, Henry Howard-Jenkins, Huixuan Tang, Hyo Jin Kim, Jaime Rivera, Ji Luo, Jing Dong, Julian Straub, Kevin Bailey, Kevin Eckenhoff, Lingni Ma, Luis Pesqueira, Mark Schwesinger, Maurizio Monge, Nan Yang, Nick Charron, Nikhil Raina, Omkar Parkhi, Peter Borschowa, Pierre Moulon, Prince Gupta, Raul Mur-Artal, Robbie Pennington, Sachin Kulkarni, Sagar Miglani, Santosh Gondi, Saransh Solanki, Sean Diener, Shangyi Cheng, Simon Green, Steve Saarinen, Suvam Patra, Tassos Mourikis, Thomas Whelan, Tripti Singh, Vasileios Balntas, Vijay Baiyya, Wilson Dreewes, Xiaqing Pan, Yang Lou, Yipu Zhao, Yusuf Mansour, Yuyang Zou, Zhaoyang Lv, Zijian Wang, Mingfei Yan, Carl Ren, Renzo De

Nardi, and Richard Newcombe. Project Aria: A New Tool for Egocentric Multi-Modal AI Research, 2023.

[93] Esri. Introduction to Ortho Mapping - ArcGIS Pro documentation. https://pro.arcgis.com/en/pro-app/latest/help/data/imagery/introduction-to-ortho-mapping.htm, 2023.

[94] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelhagen. Uncertainty-aware Vision-based Metric Cross-view Geolocalization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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

[96] Pierre Fite-Georgel, Selim Benhimane, and Nassir Navab. A Unified Approach Combining Photometric and Geometric Information for Pose Estimation. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2008.

[97] Georgios Floros, Benito van der Zander, and Bastian Leibe. OpenStreetSLAM: Global vehicle localization using OpenStreetMaps. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2013.

[98] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, 1987.

[99] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building Rome on a cloudless day. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010.

[100] Lanke Frank Tarimo Fu and Maurice Fallon. Batch Differentiable Pose Refinement for In-The-Wild Camera/LiDAR Extrinsic Calibration. In *Proc. Conf. on Robot Learning (CoRL)*, 2023.

[101] Dorian Galvez-López and Juan D. Tardos. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. on Robotics*, 28(5):1188–1197, 2012.

[102] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013.

[103] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schönberger, and Marc Pollefeys. Privacy Preserving Structure-from-Motion. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[104] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schonberger, and Marc Pollefeys. Privacy Preserving Localization and Mapping from Uncalibrated Cameras. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[105] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2DNet: Learning Accurate Correspondences for Sparse-to-Dense Feature Matching. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[106] Hugo Germain, Vincent Lepetit, and Guillaume Bourmaud. Neural Reprojection Error: Merging Feature Learning and Camera Pose Estimation. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[107] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[108] Venu Madhav Govindu. Combining Two-view Constraints for Motion Estimation. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[109] Venu Madhav Govindu. Lie-Algebraic Averaging for Globally Consistent Motion Estimation. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[110] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks.

In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[111] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[112] Chengcheng Guo, Minjie Lin, Heyang Guo, Pengpeng Liang, and Erkang Cheng. Coarse-to-fine Semantic Localization with HD Map for Autonomous Driving in Structural Scenes. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[113] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*. Wiley, 1986.

[114] Bert M. Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision (IJCV)*, 13(3):331–356, 1994.

[115] Adam W Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-BEV: What Really Matters for Multi-Sensor BEV Perception? In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[116] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, 1988.

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

[118] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[119] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[120] Suining He and S.-H. Gary Chan. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Communications Surveys Tutorials*, 2016.

[121] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. OnePose++: Keypoint-Free One-Shot Object Pose Estimation without CAD Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[122] Gim Hee Lee, Bo Li, Marc Pollefeys, and Friedrich Fraundorfer. Minimal solutions for pose estimation of a multi-camera system. In *International Journal of Robotics Research (IJRR)*, pages 521–538. Springer, 2016.

[123] Jared Heinly, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the World* in Six Days *(as Captured by the Yahoo 100 Million Image Dataset). In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[124] Joao F Henriques and Andrea Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[125] Heiko Hirschmüller, Peter R Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision (IJCV)*, 2002.

[126] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent-Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. BOP: Benchmark for 6D object pose estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[127] Yannick Hold-Geoffroy, Dominique Piché-Meunier, Kalyan Sunkavalli, Jean-Charles Bazin, François Rameau, and Jean-François Lalonde. A Deep Perceptual Measure for Lens and Camera Calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2022.

[128] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

[129] Ziyang Hong, Yvan Petillot, David Lane, Yishu Miao, and Sen Wang. TextPlace: Visual place recognition and topological localization through reading scene texts. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[130] Henry Howard-Jenkins, Jose-Raul Ruiz-Sarmiento, and Victor Adrian Prisacariu. Lalaloc: Latent layout localisation in dynamic, unvisited environments. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[131] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Koring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[132] Danying Hu, Daniel DeTone, and Tomasz Malisiewicz. Deep ChArUco: Dark ChArUco Marker Pose Estimation. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[133] Sixing Hu, Mengdan Feng, Rang M. H. Nguyen, and Gim Hee Lee. CVM-Net: Cross-View Matching Network for Image-Based Ground-to-Aerial Geo-Localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[134] Andres Huertas and Gerard Medioni. Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1986.

[135] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, and Gabriela Csurka. Robust image retrieval-based visual localization using Kapture. *arXiv:2007.13867*, 2020.

[136] Janghun Hyeon, Joohyung Kim, and Nakju Doh. Pose Correction for Highly Accurate Visual Localization in Large-Scale Indoor Spaces. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[137] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation

with deep networks. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[138] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[139] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[140] Yekeun Jeong, David Nister, Drew Steedly, Richard Szeliski, and In-So Kweon. Pushing the envelope of modern methods for bundle adjustment. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2011.

[141] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. COTR: Correspondence Transformer for Matching Across Images. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[142] Linyi Jin, Jianming Zhang, Yannick Hold-Geoffroy, Oliver Wang, Kevin Matzen, Matthew Sticha, and David F. Fouhey. Perspective Fields for Single Image Camera Calibration. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[143] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiří Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching across Wide Baselines: From Paper to Practice. *International Journal of Computer Vision (IJCV)*, 2021.

[144] Edward Johns and Guang-Zhong Yang. Feature Co-occurrence Maps: Appearance-based localisation throughout the day. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2013.

[145] Eskil Jörgensen, Christopher Zach, and Fredrik Kahl. Monocular 3D Object Detection and Box Fitting Trained End-to-End Using Intersection-over-Union Loss. *arXiv:1906.08070*, 2019.

[146] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 1976.

[147] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[148] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[149] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2015.

[150] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual SLAM for RGB-D cameras. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[151] Ali Khalajmehrabadi, Nikolaos Gatsis, and David Akopian. Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges, 2016.

[152] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proc. of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[153] Bryan Klingner, David Martin, and James Roseborough. Street View Motion-from-Structure-from-Motion. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2013.

[154] 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 *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[155] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[156] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[157] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2013.

[158] Shamit Lal, Mihir Prabhudesai, Ishita Mediratta, Adam W Harley, and Katerina Fragkiadaki. CoCoNets: Continuous Contrastive 3D Scene Representations. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[159] Dong Lao, Alex Wong, and Stefano Soatto. Does Monocular Depth Estimation Provide Better Pre-training than Classification for Semantic Segmentation? *arXiv:2203.13987*, 2022.

[160] Christos Laoudias, Michalis P. Michaelides, and Christos G. Panayiotou. Fault detection and mitigation in WLAN RSS fingerprint-based positioning. *Journal of Location Based Services*, 2012.

[161] Måns Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[162] Viktor Larsson, Marc Pollefeys, and Magnus Oskarsson. Orthographic-Perspective Epipolar Geometry. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[163] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proc. of the International Conf. on Computer Vision (ICCV) Workshops*, 2017.

[164] Karel Lebeda, Juan E. Sala Matas, and Ondřej Chum. Fixing the Locally Optimized RANSAC. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2012.

[165] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural computation*, 1989.

[166] Donghwan Lee, Soohyun Ryu, Suyong Yeon, Yonghan Lee, Deokhwa Kim, Cheolho Han, Yohann Cabon, Philippe Weinzaepfel, Nicolas Guérin, Gabriela Csurka, and Martin Humenberger. Large-scale Localization Datasets in Crowded Indoor Spaces. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[167] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

[168] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical Scene Coordinate Classification and Regression for Visual Localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[169] Xiaotian Li, Juha Ylioinas, Jakob Verbeek, and Juho Kannala. Scene Coordinate Regression with Angle-Based Reprojection Loss for Camera Relocalization. In *Proc. of the European Conf. on Computer Vision (ECCV) Workshops*, 2018.

[170] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-Resolution Correspondence Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[171] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3D point clouds. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012.

[172] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[173] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning Bird's-Eye-View Representation From Multi-Camera Images Via Spatiotemporal Transformers. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[174] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[175] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[176] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[177] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2023.

[178] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.

[179] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2D-3D matching for camera localization in a large-scale 3D map. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2017.

[180] Amy K Lobben. Navigational map reading: Predicting performance and identifying relative influence of map-related abilities. *Annals of the association of American geographers*, 97(1):64–85, 2007.

[181] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[182] Manuel Lopez, Roger Mari, Pau Gargallo, Yubin Kuang, Javier Gonzalez-Jimenez, and Gloria Haro. Deep single image camera calibration with radial distortion. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[183] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[184] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. of the International Joint Conf. on Artificial Intelligence (IJCAI)*, 1981.

[185] 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 *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[186] Zhaoyang Lv, Frank Dellaert, James M. Rehg, and Andreas Geiger. Taking a deeper look at the inverse compositional algorithm. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[187] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual-inertial localization revisited. *International Journal of Robotics Research (IJRR)*, 39(9):1061–1084, 2020.

[188] Wei-Chiu Ma, Ignacio Tartavull, Ioan Andrei Bârsan, Shenlong Wang, Min Bai, Gellert Mattyus, Namdar Homayounfar, Shrinidhi Kowshika Lakshmikanth, Andrei Pokrovsky, and Raquel Urtasun. Exploiting sparse semantic HD maps for self-driving vehicle localization. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[189] Wei-Chiu Ma, Shenlong Wang, Marcus A. Brubaker, Sanja Fidler, and Raquel Urtasun. Find your way by observing the sun and other semantic cues. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2017.

[190] Wei-Chiu Ma, Shenlong Wang, Jiayuan Gu, Sivabalan Manivasagam, Antonio Torralba, and Raquel Urtasun. Deep Feedback Inverse Problem Solver. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[191] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[192] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. *Methods for Non-Linear Least Squares Problems*. 2004.

[193] F. Landis Markley, Yang Cheng, John L. Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[194] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[195] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[196] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip H. S. Torr. Random Forests versus Neural Networks - What's Best for Camera Localization? In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2017.

[197] Lili Meng, Jianhui Chen, Frederick Tung, James J Little, Julien Valentin, and Clarence W de Silva. Backtracking regression forests for accurate camera relocalization. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[198] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-DOF localization on mobile devices. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014.

[199] Krystian Mikolajczyk and Cordelia Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision (IJCV)*, 2004.

[200] 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 *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[201] Michael J. Milford and Gordon. F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2012.

[202] Zhixiang Min, Naji Khosravan, Zachary Bessinger, Manjunath Narayana, Sing Bing Kang, Enrique Dunn, and Ivaylo Boyadzhiev. LASER: LAtent SpacE Rendering for 2D Visual Localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[203] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiří Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[204] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[205] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

[206] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Real Time Localization and 3D Reconstruction. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[207] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. on Graphics (ToG)*, 2022.

[208] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-End 3D Scene Reconstruction from Posed Images. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[209] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-DoF global localization in outdoor environments. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[210] Tony Ng, Hyo Jin Kim, Vincent T. Lee, Daniel DeTone, Tsun-Yi Yang, Tianwei Shen, Eddy Ilg, Vassileios Balntas, Krystian Mikolajczyk, and Chris Sweeney. NinjaDesc: Content-Concealing Visual Descriptors via Adversarial

Learning. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[211] Tony Ng, Adrian Lopez-Rodriguez, Vassileios Balntas, and Krystian Mikolajczyk. Reassessing the Limitations of CNN Methods for Camera Pose Regression. *arXiv:2108.07260*, 2021.

[212] David Nistér. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2004.

[213] David Nistér, Oleg Naroditsky, and James Bergen. Visual Odometry. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[214] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-Scale Image Retrieval with Attentive Deep Local Features. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[215] John O'Keefe and Lynn Nadel. *The hippocampus as a cognitive map*. Clarendon Press, 1978.

[216] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning Local Features from Images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[217] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint*, 2018.

[218] OpenStreetMap contributors. OpenStreetMap: The Free Wiki World Map. `https://www.openstreetmap.org`, 2017.

[219] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[220] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2009.

[221] Nikhil R Pal and Sankar K Pal. A review on image segmentation techniques. *Pattern recognition*, 1993.

[222] Pilailuck Panphattarasap and Andrew Calway. Automated map reading: image based localisation in 2-D maps using binary semantic descriptors. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[223] Seonwook Park, Thomas Schöps, and Marc Pollefeys. Illumination Change Robustness in Direct Visual SLAM. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2017.

[224] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[225] Jan-Hendrik Pauls, Kürsat Petek, Fabian Poggenhans, and Christoph Stiller. Monocular Localization in HD Maps by Combining Semantic Segmentation and Distance Transform. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[226] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online Invariance Selection for Local Feature Descriptors. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[227] Rong Peng and Mihail L. Sichitiu. Angle of Arrival Localization for Wireless Sensor Networks. In *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, 2006.

[228] Shuxue Peng, Zihang He, Haotian Zhang, Ran Yan, Chuting Wang, Qingtian

Zhu, and Xiao Liu. MegLoc: A Robust and Accurate Visual Localization Pipeline. *arXiv preprint*, 2021.

[229] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional Occupancy Networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[230] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[231] Jonah Philion and Sanja Fidler. Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[232] Pedro O. Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron Courville. Unsupervised Learning of Dense Visual Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[233] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. Benchmarking Image Retrieval for Visual Localization. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2020.

[234] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing Scenes by Inverting Structure from Motion Reconstructions. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[235] Robert Pless. Using many cameras as one. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[236] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual Modeling with a Hand-held Camera. *International Journal of Computer Vision (IJCV)*, 2004.

[237] Filip Radenovic, Johannes L Schonberger, Dinghuang Ji, Jan-Michael Frahm, Ondrej Chum, and Jiri Matas. From dusk till dawn: Modeling in the dark. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[238] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2018.

[239] Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):4407–4414, 2018.

[240] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[241] Anita Rau, Guillermo Garcia-Hernando, Danail Stoyanov, Gabriel J. Brostow, and Daniyar Turmukhambetov. Predicting Visual Overlap of Images Through Interpretable Non-Metric Box Embeddings. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[242] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[243] Santiago Cortés Reina, Arno Solin, Esa Rahtu, and Juho Kannala. ADVIO: An authentic dataset for visual-inertial odometry. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[244] Jérôme Revaud, Jon Almazán, Rafael Sampaio de Rezende, and César Roberto de Souza. Learning With Average Precision: Training Image Retrieval With a Listwise Loss. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[245] Jérôme Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: Repeatable and Reliable Detector and Descriptor. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[246] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient Neighbourhood Consensus Networks via Submanifold Sparse Convolutions. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[247] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[248] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[249] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic Feature Transform for Monocular 3D Object Detection. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2019.

[250] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[251] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006.

[252] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006.

[253] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2011.

[254] Philipp Ruchti, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Localization on OpenStreetMap data using a 3D laser scanner. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2015.

[255] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Proc. of the International Conf. on 3D Digital Imaging and Modeling (3DIM)*, 2001.

[256] Chris Russell, Matteo Toso, and Neill Campbell. Fixing Implicit Derivatives: Trust-Region Based Learning of Continuous Energy Functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[257] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating Images into Maps. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2022.

[258] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey

Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[259] Noe Samano, Mengjie Zhou, and Andrew Calway. You are here: Geolocation by embedding maps and images. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[260] Paul-Edouard Sarlin. Visual localization made easy with hloc. `https://github.com/cvg/Hierarchical-Localization/`.

[261] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[262] Paul-Edouard Sarlin, Frédéric Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging Deep Visual Descriptors for Hierarchical Efficient Localization. In *Proc. Conf. on Robot Learning (CoRL)*, 2018.

[263] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[264] Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Bulo, Richard Newcombe, Peter Kontschieder, and Vasileios Balntas. OrienterNet: Visual Localization in 2D Public Maps with Neural Matching. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[265] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[266] Paul-Edouard Sarlin, Philipp Lindenberger, Viktor Larsson, and Marc Polle-

feys. Pixel-Perfect Structure-From-Motion With Featuremetric Refinement. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2023.

[267] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. SNAP: Self-Supervised Neural Maps for Visual Positioning and Semantic Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[268] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Victor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[269] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-Scale Location Recognition And The Geometric Burstiness Problem. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[270] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast Image-Based Localization using Direct 2D-to-3D Matching. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2011.

[271] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving Image-Based Localization by Active Correspondence Search. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012.

[272] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 39(9):1744–1756, 2016.

[273] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[274] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image

Retrieval for Image-Based Localization Revisited. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2012.

[275] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of CNN-based absolute camera pose regression. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[276] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. SimpleRecon: 3D reconstruction without 3D convolutions. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[277] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 2002.

[278] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[279] Johannes Lutz Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[280] Johannes Lutz Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[281] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016.

[282] Thomas Schops, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle Adjusted Direct RGB-D SLAM. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[283] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo

benchmark with high-resolution images and multi-camera videos. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[284] Prafull Sharma, Ayush Tewari, Yilun Du, Sergey Zakharov, Rares Ambrus, Adrien Gaidon, William T Freeman, Fredo Durand, Joshua B Tenenbaum, and Vincent Sitzmann. Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[285] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. RANSAC-Flow: generic two-stage image alignment. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[286] Yujiao Shi and Hongdong Li. Beyond Cross-view Image Retrieval: Highly Accurate Vehicle Localization Using Satellite Image. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[287] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-Aware Feature Aggregation for Image based Cross-View Geo-Localization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[288] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am I looking at? Joint Location and Orientation Estimation by Cross-View Matching. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[289] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. Optimal feature transport for cross-view image geo-localization. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2020.

[290] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy preserving visual SLAM. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[291] 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 *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[292] Reid G. Simmons and Sven Koenig. Probabilistic Robot Navigation in Partially Observable Environments. In *Proc. of the International Joint Conf. on Artificial Intelligence (IJCAI)*, 1995.

[293] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[294] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo Tourism: exploring photo collections in 3D. In *ACM Trans. on Graphics (TOG)*, 2006.

[295] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. In *International Journal of Computer Vision (IJCV)*, 2008.

[296] Sergei Solonets, Daniil Sinitsyn, Lukas Von Stumberg, Nikita Araslanov, and Daniel Cremers. An analytical solution to gauss-newton loss for direct image alignment. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2024.

[297] Pablo Speciale, Johannes L. Schönberger, Sing Bing Kang, Sudipta N. Sinha, and Marc Pollefeys. Privacy Preserving Image-Based Localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[298] Pablo Speciale, Johannes L. Schönberger, Sudipta N. Sinha, and Marc Pollefeys. Privacy Preserving Image Queries for Camera Localization. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[299] Pablo Speciale, Johannes Lutz Schönberger, Sing Bing Kang, Sudipta N Sinha, and Marc Pollefeys. Privacy Preserving Image-Based Localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[300] Jaime Spencer, Richard Bowden, and Simon Hadfield. Same Features, Different Day: Weakly Supervised Feature Learning for Seasonal Invariance. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[301] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)*, 2014.

[302] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[303] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[304] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[305] Xun Sun, Yuanfan Xie, Pei Luo, and Liang Wang. A dataset for benchmarking image-based localization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[306] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[307] Chris Sweeney, John Flynn, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. Efficient Computation of Absolute Pose for Gravity-Aware Augmented Reality. In *Proc. of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2015.

[308] 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. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2019.

[309] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[310] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-

NeRF: Scalable Large Scene Neural View Synthesis. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[311] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment network. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019.

[312] Jiexiong Tang, Hanme Kim, Vitor Guizilini, Sudeep Pillai, and Rares Ambrus. Neural Outlier Rejection for Self-Supervised Keypoint Learning. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.

[313] Tim Y. Tang, Daniele De Martini, Shangzhe Wu, and Paul Newman. Self-supervised learning for using overhead imagery as maps in outdoor range sensor localization. *International Journal of Robotics Research (IJRR)*, 40(12-14):1488–1509, 2021.

[314] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[315] Zachary Teed, Lahav Lipson, and Jia Deng. Deep Patch Visual Odometry. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[316] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What Makes for Good Views for Contrastive Learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[317] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second Order Similarity Regularization for Local Descriptor Learning. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[318] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, Fredrik Kahl, and Torsten Sattler. Long-Term Visual Localization Revisited. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2020.

[319] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic Match Consistency for Long-

Term Visual Localization. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[320] Engin Tola, Vincent Lepetit, and Pascal Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2009.

[321] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2013.

[322] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[323] Akihiko Torii, Hajime Taira, Josef Sivic, Marc Pollefeys, Masatoshi Okutomi, Tomas Pajdla, and Torsten Sattler. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2019.

[324] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment — a modern synthesis. In *International workshop on vision algorithms*, 1999.

[325] Gabriele Trivigno, Carlo Masone, Barbara Caputo, and Torsten Sattler. The Unreasonable Effectiveness of Pre-Trained Features for Camera Pose Refinement. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[326] Eduard Trulls, Yuhe Jin, Kwang Moo Yi, Dmytro Mishkin, Jiri Matas, and Pascal Fua. CVPR 2020 Image Matching Challenge. `https://www.cs.ubc.ca/research/image-matching-challenge/`. Accessed March 1, 2021.

[327] Prune Truong, Martin Danelljan, and Radu Timofte. GLU-Net: Global-Local Universal Network for dense flow and correspondences. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[328] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning Accurate Dense Correspondences and When To Trust Them. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[329] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[330] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[331] Michał J Tyszkiewicz, Kevis-Kokitsi Maninis, Stefan Popov, and Vittorio Ferrari. RayTran: 3D pose estimation and shape reconstruction of multiple objects from videos with ray-traced transformers. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[332] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1991.

[333] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Depth and Motion Network for Learning Monocular Stereo. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[334] Dorin Ungureanu, Federica Bogo, Silvano Galliani, Pooja Sama, Xin Duan, Casey Meekhof, Jan Stühmer, Thomas J. Cashman, Bugra Tekin, Johannes L. Schönberger, Pawel Olszta, and Marc Pollefeys. HoloLens 2 Research Mode as a Tool for Computer Vision Research, 2020.

[335] Julien Valentin, Matthias Niessner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip H. S. Torr. Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[336] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 2008.

[337] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Single-image Calibration with Geometric Optimization. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024.

[338] Tomas Vojir, Ignas Budvytis, and Roberto Cipolla. Efficient Large-Scale Semantic Visual Localization in 2D Maps. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2020.

[339] Lukas Von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. GN-Net: The Gauss-Newton loss for multi-weather relocalization. *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[340] Lukas Von Stumberg, Patrick Wenzel, Nan Yang, and Daniel Cremers. LM-Reloc: Levenberg-Marquardt Based Direct Visual Relocalization. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2020.

[341] Olga Vysotska and Cyrill Stachniss. Improving SLAM by exploiting building information from publicly available maps and localization priors. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 85(1):53–65, 2017.

[342] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using LSTMs for structured feature correlation. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2017.

[343] Johanna Wald, Torsten Sattler, Stuart Gododetz, Tommaso Cavallari, and Federico Tombari. Beyond Controlled Environments: 3D Camera Re-Localization in Changing Indoor Scenes. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[344] Chaoyang Wang, Hamed Kiani Galoogahi, Chen-Hsuan Lin, and Simon Lucey. Deep-LK for efficient adaptive object tracking. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2018.

[345] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. PatchmatchNet: Learned Multi-View Patchmatch Stereo. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[346] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: Solving Pose Estimation via Diffusion-aided Bundle Adjustment. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2023.

[347] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning Multi-View Image-Based Rendering. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[348] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning Feature Descriptors using Camera Pose Supervision. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[349] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. MatchFormer: Interleaving Attention in Transformers for Feature Matching. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2022.

[350] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, and Juho Kannala. Continual Learning for Image-Based Camera Localization. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[351] Li Weng, Valérie Gouet-Brunet, and Bahman Soheilian. Semantic signatures for large-scale visual localization. *Multimedia Tools and Applications*, 2021.

[352] Patrick Wenzel, Rui Wang, Nan Yang, Qing Cheng, Qadeer Khan, Lukas von Stumberg, Niclas Zeller, and Daniel Cremers. 4Seasons: A Cross-Season Dataset for Multi-Weather SLAM in Autonomous Driving. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2020.

[353] Kyle Wilson and Noah Snavely. Robust Global Translations with 1DSfM. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014.

[354] Oliver J. Woodford and Edward Rosten. Large scale photometric bundle adjustment. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2020.

[355] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krahenbuhl. Sampling Matters in Deep Embedding Learning. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2017.

[356] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian FP Kooij. Visual Cross-View Metric Localization with Dense Uncertainty Estimates. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[357] Binbin Xu, Andrew J. Davison, and Stefan Leutenegger. Deep Probabilistic Feature-Metric Tracking. *IEEE Robotics and Automation Letters (RA-L)*, 6(1):223–230, 2021.

[358] Fan Yan, Olga Vysotska, and Cyrill Stachniss. Global Localization on OpenStreetMap Using 4-bit Semantic Descriptors. In *Proc. European Conf. on Mobile Robotics (ECMR)*, 2019.

[359] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1127–1134, 2020.

[360] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. SANet: Scene agnostic network for camera localization. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[361] Tsun-Yi Yang, Duy-Kien Nguyen, Huub Heijnen, and Vassileios Balntas. UR2KiD: Unifying Retrieval, Keypoint Detection, and Keypoint Description without Local Correspondence Supervision. *arXiv:2001.07252*, 2020.

[362] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth Inference for Unstructured Multi-view Stereo. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[363] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent MVSNet for high-resolution multi-view stereo depth inference. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[364] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned Invariant Feature Transform. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016.

[365] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to Find Good Correspondences. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[366] Baosheng Yu and Dacheng Tao. Heatmap Regression via Randomized Rounding. *arXiv:2009.00225*, 2020.

[367] Zehao Yu and Shenghua Gao. Fast-MVSNet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[368] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera Pose Voting for Large-Scale Image-Based Localization. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2015.

[369] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[370] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose: Predicting Probabilistic Relative Rotation for Single Objects in the Wild. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[371] 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. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2019.

[372] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference Pose Generation for Long-term Visual Localization via Learned Features and View Synthesis. *International Journal of Computer Vision (IJCV)*, 2021.

[373] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive Correspondence Pruning by Consensus Learning. In *Proc. of the International Conf. on Computer Vision (ICCV)*, 2021.

[374] Mengjie Zhou, Xieyuanli Chen, Noe Samano, Cyrill Stachniss, and Andrew Calway. Efficient Localisation Using Images and OpenStreetMaps. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[375] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847*, 2018.

[376] Qunjie Zhou, Sergio Agostinho, Aljosa Osep, and Laura Leal-Taixe. Is Geometry Enough for Matching in Visual Localization? In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[377] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2Pix: Epipolar-Guided Pixel-Level Correspondences. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[378] Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To Learn or Not to Learn: Visual Localization from Essential Matrices. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2020.

[379] Sijie Zhu, Taojiannan Yang, and Chen Chen. VIGOR: Cross-view image geo-localization beyond one-to-one retrieval. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[380] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Acronyms

2D 2-dimensional
3D 3-dimensional
AR Augmented Reality
AUC area under the curve
BA bundle adjustment
BEV Bird's-Eye View
CNN Convolutional Neural Network
DNN Deep Neural Network
DoF Degree of Freedom
FoV field of view
GN Gauss-Newton
GNSS Global Navigation Satellite System
GT ground truth
ICP Iterative Closest Point
KA keypoint adjustment
LiDAR Light Detection And Ranging
LM Levenberg-Marquardt
MVS Multi-View Stereo
OSM OpenStreetMap
PGO pose graph optimization
PnP Perspective-n-Point
RANSAC Random Sample Consensus
SfM Structure-from-Motion
SIFT Scale-Invariant Feature Transform
SLAM Simultaneous Localization and Mapping