BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

VISUAL LOCALIZATION IN NATURAL ENVIRONMENTS

VIZUÁLNÍ LOKALIZACE V PŘÍRODĚ

PHD THESIS

DISERTAČNÍ PRÁCE

AUTHOR **Ing. JAN BREJCHA**

AUTOR PRÁCE

SUPERVISOR **Doc. Ing. MARTIN ČADÍK, Ph.D.**

ŠKOLITEL

BRNO 2021

Abstract

We focus our work on camera position and orientation estimation given a query photograph; we call this problem *visual geo-localization*. Specifically, we focus on photographs captured in natural, mountainous environments. We introduce a thorough review of state-of-the-art computer vision methods, datasets, and evaluation practices for visual geo-localization problems. The survey revealed that researchers usually cast visual geo-localization in natural environments as a similarity or a correspondence search between an input photograph and a terrain model; we call this problem the *cross-domain matching*. We identified three main goals to improve over the state of the art in visual geo-localization in mountainous environments using cross-domain matching: (I) the need for new datasets for training, validation, and evaluation of cross-domain visual geo-localization algorithms, (II) the need to verify whether the cross-domain matching algorithms may benefit from using different features—horizon lines, edge maps, semantic segmentation, and satellite imagery, (III) the need to illustrate the usefulness of visual geo-localization methods by developing novel applications.

In this thesis, we thoroughly describe our research studies to illustrate how we examined particular goals. We introduce several novel datasets for evaluation and training of cross-domain matching methods. These novel datasets allowed us to propose a novel method for cross-domain photo-to-terrain matching using a combination of semantic segments and classic edge-based features. We illustrate the benefits of our novel approach over the state of the art on camera orientation estimation. Furthermore, we propose a meta-algorithm based on a *cross-domain Structure from Motion* for a weakly supervised acquisition of cameras aligned with the synthetic terrain. This novel cross-domain data acquisition scheme allowed us to train a compact cross-domain keypoint descriptor. We illustrate the descriptor performance by estimating full camera pose by matching the query photograph to the rendered terrain model.

Finally, we demonstrate a practical usability of outdoor visual geo-localization by designing a novel application of photography presentation on a computer screen or in virtual reality. Moreover, we illustrate that our novel presentation method helps the user with complex outdoor scene understanding and improves self-localization in unvisited outdoor environments.

Abstrakt

V této práci se zabýváme odhadem pozice a orientace kamery z dané fotografie. Tento problém nazýváme *vizuální geo-lokalizace*. Konkrétně se zabýváme fotografiemi pořízenými v přírodních horských prostředích. Představujeme podrobný průzkum aktuálního stavu poznání algoritmů, datových sad a přístupů k vyhodnocování problému vizuální geo-lokalizace. Náš průzkum odhalil, že vizuální geo-lokalizace v přírodních prostředích je často řešena pomocí vyhledávání podobností nebo korespondencí mezi vstupní fotografií a terénním modelem. Problém nacházení korespondencí mezi fotografií a terénním modelem nazýváme *porovnávání napříč doménami* (cross-domain matching). Na základě našeho průzkumu jsme stanovili tři hlavní cíle, jejichž dosažení nám umožňuje překonat aktuální stav poznání vizuální geo-lokalizace v horských prostředích s využitím porovnávání napříč doménami: (I) potřeba nových datových sad které umožní trénovat, vyhodnocovat a porovnávat algoritmy vizuální geo-lokalizace, (II) potřeba ověřit, zda využití různých příznaků – křivek horizontu, hranových map, sémantické segmentace a satelitních snímků pomůže vylepšit algoritmy pro porovnávání napříč doménami, (III) potřeba ilustrovat využitelnost metod vizuální geo-lokalizace pomocí vývoje jejich nových aplikací.

V této práci podrobně popisujeme naše výzkumné studie, které objasňují, jakým způsobem jsme postupovali ve výzkumu jednotlivých cílů. Představujeme několik nových datových sad pro účely vyhodnocování, porovnávání a trénování jednotlivých metod. S využitím těchto nových datových sad jsme vyvinuli novou metodu pro zarovnání fotografií s terénním modelem na základě sémantické segmentace kombinované s běžnými hranovými příznaky. Pomocí experimentálního vyhodnocení objasňujeme výhody našeho nového přístupu oproti aktuálnímu stavu poznání. Dále navrhujeme meta algoritmus umožňující automatickou kalibraci více kamer, který je založen na odhadu *struktury z pohybu* (Structure from Motion) *napříč doménami*. Tento nový přístup pro automatické zarovnávání fotografií s terénním modelem nám umožňuje natrénovat kompaktní deskriptor klíčových bodů pomocí hlubokého učení. V rámci našeho výzkumu ukazujeme funkčnost tohoto deskriptoru při odhadu externích parametrů kamery (pozice a orientace) pomocí porovnávání vstupní fotografie s terénním modelem.

V závěru práce ukazujeme praktickou využitelnost našich metod pro automatickou kalibraci externích parametrů kamery. Navrhujeme nový přístup k prezentaci fotografií, který je vhodný jak pro prezentaci na monitoru či jiné projekční ploše, tak pro virtuální realitu. Pomocí experimentálního vyhodnocení ukazujeme, že naše nová metoda prezentace fotografií pomáhá uživatelům s orientací v neznámých komplexních přírodních scénách.

Keywords

Visual geo-localization, camera localization, camera rotation estimation, digital elevation models, terrain rendering, cross-domain matching, descriptor matching, photography presentation, virtual reality, augmented reality.

Klíčová slova

Vizuální geo-lokalizace, lokalizace kamery, odhad rotace kamery, digitální elevační modely, renderování terénu, porovnávání napříč doménami, porovnávání deskriptorů, prezentace fotografií, virtuální realita, rozšířená realita.

Reference

BREJCHA, Jan. *Visual Localization in Natural Environments*. Brno, 2021. PhD thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Doc. Ing. Martin Čadík, Ph.D.

Visual Localization in Natural Environments

Declaration

I hereby declare that this dissertation thesis was prepared as an original work by the author under the supervision of Martin Čadík. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

....

Jan Brejcha
January 14, 2021

Acknowledgements

In the first place, I would like to thank my supervisor, Martin Čadík, for excellent six-year cooperation, countless consultations, for being not only a fantastic leader but also a good friend. I am indebted to my wife, who supported me all those years, even in situations that were not easy to bear. I want to thank all my colleagues from the department of computer graphics and multimedia who created a great atmosphere and were always there to discuss all my ideas and to enjoy a cup of coffee. Many thanks also to my colleagues at Adobe Research, who tremendously helped me with two research projects and papers contained in this thesis. I also would like to thank Martin Šimonovský for many inspiring talks and for his contribution of Weighted Edge Detector (See 4.2.2) in the early stage of my Ph.D. journey. Finally, I want to thank the whole Brno University of Technology, Faculty of Information Technology, which supported me materially and financially, which allowed me to work on my topic.

To the end of material and financial support, this work was supported by several projects and agencies, namely by project no. LTAIZ19004 Deep-Learning Approach to Topographical Image Analysis by the Ministry of Education, Youth and Sports of the Czech Republic within the activity INTER-EXCELENCE (LT), subactivity INTER-ACTION (LTA), ID: SMSM2019LTAIZl; SoMoPro II grant (financial contribution from the EU 7 FP People Programme Marie Curie Actions, REA 291782, and from the South Moravian Region); by the TACR Competence Centres project V3C – Visual Computing Competence Center (no. TE01020415); and by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center – LM2015070".

Contents

List of Figures

List of Tables

List of Acronyms

ALE Automatic Labeling Environment

API Appication Programming Interface

AR Augmented Reality

ASMK Aggregated Selective Match Kernel

BOW Bag Of Words

CNN Convolutional Neural Network

DEM Digital Elevation Model

DOF Degrees Of Freedom

DP Dynamic Programming

ECEF Earth-centerd, Earth-fixed

FCN Fully Convolutional Networks

FOV Field Of View

FREAK Fast Retina Keypoint

GIS Geographic Information System

GMCP Generalized Minimum Clique Problem

GPS Global Positioning System

ICP Iterative Closest Points

INS Inertial Navigation System

IR Infrared

LIDAR Light Detection And Ranging

LMeds Least Median Of Squares

MST Minimum Spanning Tree

MVS Multi-view Stereo

PnP Perspective-*n*-Point

RANSAC Random Sample Consensus

SARE Stochastic Attraction And Repulsion Embedding Loss Function

SfM Structure From Motion

SIFT Scale Invariant Feature Transform

SLAM Simultaneous Localization And Mapping

SURF Speeded-Up Robust Features

SVM Support Vector Machine

UAV Unmanned Areal Vehicle

V-SLAM Visual Simultaneous Localization And Mapping

VLAD Vector Of Locally Agreggated Descriptors

VR Virtual Reality

WGS84 World Geodetic System

Chapter 1

Introduction

Billions of images and videos on the Internet comprise a large amount of valuable information covering ever-growing geographic areas. However, despite the proliferation of cameras and mobile devices equipped with Global Positioning System (GPS) sensor, the majority of available media still lack the geo-tag information; according to Flatow *et al.* [56] (2015), there was around 2% of geo-tagged media on Twitter and 25% on Instagram.

The location gives the context, and it is essential for image and video recognition. Essential applications are crucially dependent on the location knowledge, e.g., model-based image enhancement [99], augmented reality [125, 143, 15], self-driving vehicles [28, 111], and more. Additionally, visual geo-localization could help existing non-visual localization systems to achieve higher precision and robustness. These facts motivate our work to build new algorithms that would *localize* visual documents, such as photographs and videos. Let us first review some definitions of *visual localization* given by various authors in the literature.

1.1 What is Visual Geo-localization?

Hays and Efros [77] introduce visual geo-localization as "... estimating a distribution over geographic locations from single image..." Zamir and Shah [224] define the problem as "... estimating the geo-location of a query image by finding its matching reference images."[1] Bansal *et al.* [18] say, "Given a ground-level street-view image in an urban area, we want to determine the geo-location of the camera in the absence of any metadata (GPS or camera parameters)." In summary, we define visual geo-localization as searching for the geographic coordinates (and possibly the camera orientation), which captured the query image or the video frame. We will use the term *geo-localization* instead of *localization* since it is more accurate in our context. Whenever we use the term geo-localization, it is clear that we are searching across geographical locations.

The problem has several variants—with known GPS position estimate or coarse region of interest, the task is to find the precise position of the camera. An example of an existing

[1] Author's note: in a reference database of geo-tagged images.

method dealing with this problem is PoseNET [95], where the authors estimate the precise location and camera orientation with a convolutional neural network (CNN) inside a small area in Cambridge. Without a known GPS position estimate, the task is to localize the camera on a large scale. Researchers already approached large scale geo-localization at the scale of the whole Earth [77, 213, 207, 173]. However, such *global* approaches still provide only coarse geo-localization (with reasonable uncertainty at the country level, as reported by PlaNET [213]). Large scale geo-localization was therefore approached from other perspectives as well—at the region level [116], city-scale level [7, 166, 202, 149, 114, 159, 13], and at the level of natural environments, particularly mountains [69, 14, 163, 37]. Usually, there is an assumption that the camera intrinsics such as field of view (FOV) is known, but this information may not be available in many practical scenarios, making the geo-localization task even harder. The geo-localization in natural environments is a less studied variant compared to the visual geo-localization in urban areas. While methods dealing with visual geo-localization in the urban environment are precise (up to meters in the best scenarios), the resolution of the state-of-the-art visual geo-localization method in the outdoor environment [163] is one kilometer.

1.2 Selected Problems

Already being said, visual geo-localization in outdoor natural environments is a less studied problem than visual geo-localization in urban areas. Moreover, urban areas cover approximately 1% of the Earth's surface [151]. These facts motivate us to focus our work on visual geo-localization in natural environments. Outdoor natural environments bring their unique challenges, compared to the ones posed at urban and city-scale. These challenges include but are not limited to weather and seasonal changes, terrain self-similarity, and low coverage of ground-level imagery (*e.g.*, Google StreetView is available only in most visited outdoor areas). To overcome the issue of limited ground-level imagery, we chose to pursue the goal of *cross-domain matching*—comparing real photographs with a synthetically rendered terrain counterpart. To render a synthetic terrain, we use DEMs, which are publicly available for the whole Earth[2]. Visual localization based on DEM makes sense only in non-flat environments. Therefore, we confine us to the mountainous environments, where distinctive features (e.g., mountain peaks) are available. In this thesis, we work with a plain DEM, with a DEM overlaid with natural semantic segmentation, or with a DEM overlaid with a satellite orthophoto map; see the illustration in Fig 1.1.

One of the fundamental problems is the lack of datasets with photographs precisely aligned with the terrain model. The low number and diversity of such datasets make

[2] http://viewfinderpanoramas.org

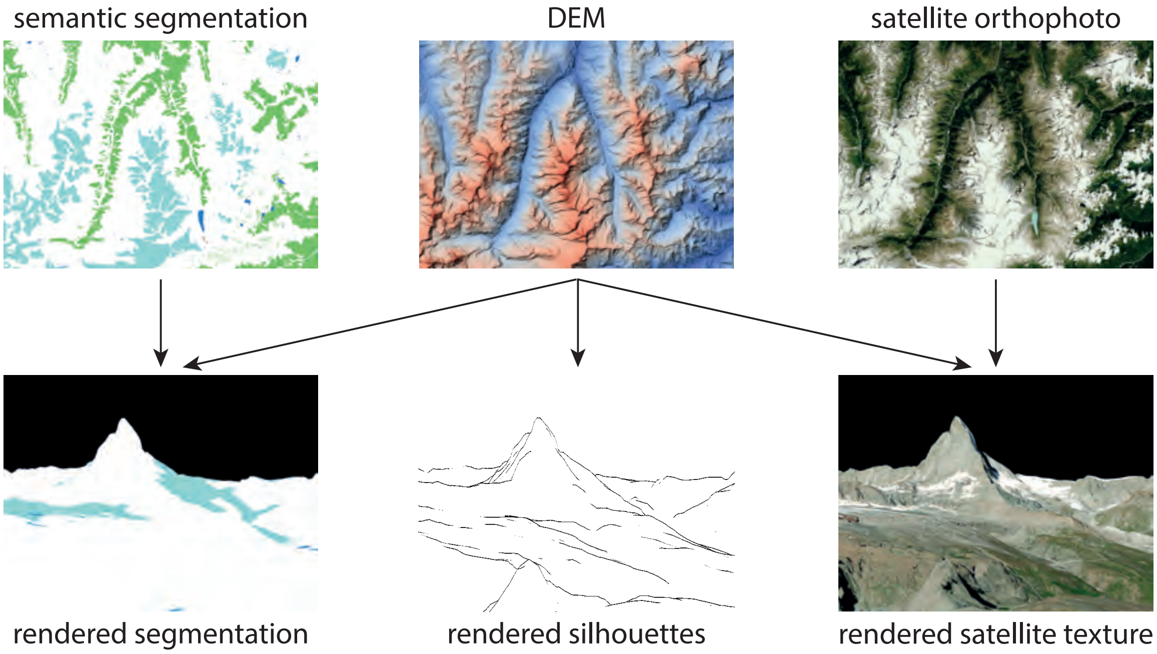


Figure 1.1: Illustration of DEM and additional overlays. Left: natural semantic segmentation (green: forests, light blue: glaciers, dark blue: water bodies); middle: the source DEM; right: satellite orthophoto texture; bottom: corresponding rendered views of Matterhorn, at the borders of Switzerland and Italy.

the development and evaluation of novel methods difficult. To foster the research in this area, we studied how to build large scale datasets for image-to-terrain matching, either automatically or with manual annotation.

The camera orientation estimation is the second problem we focus on in this thesis. With a known position and the query image, the goal is to estimate three angles of camera orientation. Estimating camera orientation is beneficial if we precisely know its position—usually using a precise GPS sensor. This thesis proposes a novel method based on comparing edges and semantic segments from the query image with synthetically rendered silhouettes (Fig. 1.1-middle) and semantic segmentation (Fig. 1.1-left).

However, the precise position of the photograph is often unknown. In this case, we need a full camera pose estimation method, which jointly estimates camera position and orientation. We focused on camera pose estimation using the Perspective-*n*-Point (PnP) algorithm. The critical problem is finding the cross-domain correspondences between the keypoints detected in the query and rendered images. To solve this problem, we trained a CNN to produce a cross-domain descriptor, a vector of numbers similar for corresponding keypoints and dissimilar for non-corresponding keypoints.

The human (in)ability to estimate a photograph's location [207] from visual data is also a related problem. Users' orienteering and localization capabilities are essential to understand image-based navigation instructions. We studied this problem on the application of photography presentation, where our goal was to help users understand the link between the presented material and its location in an enjoyable way.

1.3 Contributions

In summary, in this thesis, we present the following contributions:

1. We created an in-depth survey of existing approaches, datasets, and evaluation methods in the field of visual geo-localization [A2]. This thesis further updates this survey to contain more complete sets of methods relevant to our following contributions.

2. We introduced a novel cross-domain dataset, GeoPose3K [A1], for training and evaluating visual geo-localization methods in outdoor environments. GeoPose3K contains more than three thousand photographs aligned with a terrain model, accompanied by various metadata rendered from a terrain model, such as depth and normal maps, synthetic silhouette edges, and simulation of illumination throughout the day.

3. We examined structure from motion (SfM) methods to reconstruct and align unsorted photograph collections with the terrain model [A4]. We proposed a novel method *structure from motion with terrain reference*. Our *SfM with terrain reference* uses a terrain model to initialize the scene reconstruction, which avoids drift of the real photographs [A5].

4. We proposed a method to make camera orientation estimation by matching photograph to terrain model more robust. We jointly use the semantic segmentation with edge-based features to allow three degrees of freedom (DOF) camera orientation estimation with the apriori known camera position [A3].

5. We proposed a novel cross-domain descriptor based on a convolutional neural network (CNN) [A5]. Our novel descriptor allowed us to employ keypoint-based techniques to directly localize an input photograph relative to a rendered terrain model.

6. We proposed a novel application of visual geo-localization in outdoor environments. Our method allows users to automatically generate a fly through a virtual terrain with their photographs aligned [A4, A6]. This way, the user can re-create his hike inside a virtual reality or on a computer screen. We also experimentally show that our new mode of presentation helps users self-localize better in previously unknown environments.

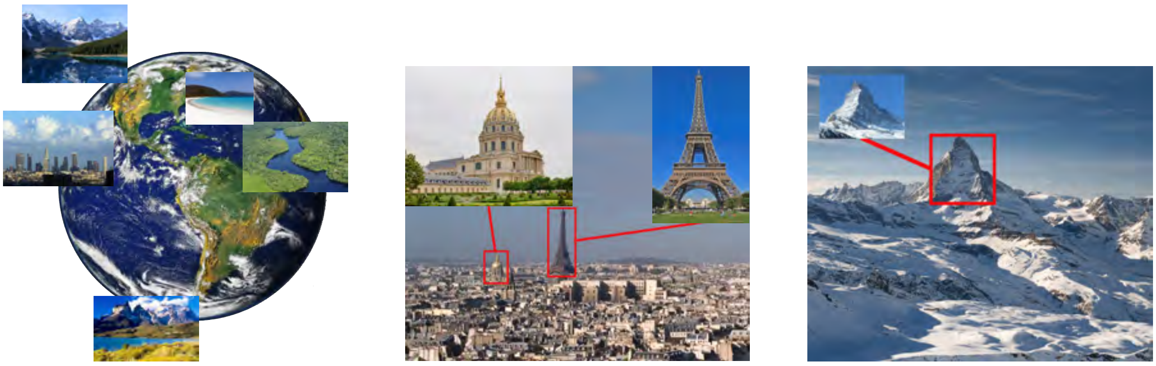
Part I

State-of-the-art in Visual Geo-localization

We can instantiate visual geo-localization in many different variants. We can categorize it according to the area: small-scale, large-scale, or around the whole planet, according to the environment, *e.g.*, urbanized areas, countryside, or mountains. Even though the approaches vary and we can see the geo-localization from many different viewpoints, the ultimate goal is the same for all of them: recovering the camera location given the visual document the camera captured. Although we focus on visual geo-localization in natural environments, let us review the visual camera geo-localization and place recognition as a whole. While geo-localization in natural environments is not in researchers' main focus yet, geo-localization (or place recognition) in other environments, particularly cities, has recently gained attention from many researchers. Besides the works dealing with visual geo-localization, we also review existing datasets and evaluation methods.

Chapter 2

A Survey of Visual Geo-localization Methods



(a) Category: global[1]

(b) Category: city-scale[2]

(c) Category: natural[3]

Figure 2.1: Illustration of visual geo-localization categories.

2.1 Classification of Visual Geo-localization Methods

We classify the works in this survey by two main criteria. The first criterion is the type of input data. We recognize two main classes of methods concerning the type of input data: *image-based* methods and *methods utilizing data of multiple modalities*. ***Image-based methods*** use large GPS-tagged image databases to infer the location of the query image. These methods can locate (up to several centimeters in some cases) images, mainly in highly urbanized areas, with a high density of ground-level imagery available online. Methods utilizing ***data of multiple modalities*** use more information than image-based methods, beyond a simple image database. Mostly, the methods make use of DEMs [14, 15, 143, 200], orthophoto maps, attribute maps [116], or satellite imagery [85]. Such methods were developed mainly for areas where coverage by ground-level imagery is sparse, *e.g.*, mountain areas, deserts, and other places with low population density.

[1] Credit: N. Palmer (CIAT)—Amazonia, M. Pazzani—Caribbean Island, T. Pintaric—Los Angeles Dowtown

[2] Credit: Myrabella—Paris from Notre Dame, Diliff—Les Invalides

[3] Credit: Felix Lamouroux—Zermatt Panorama, Marcel Wiesweg—Matterhorn

Categorization based only on the type of data would not be enough, since the categories may overlap. To distinguish between the methods better, we add a second classification criterion—the particular method's environment for which it was designed. We divide the environment criterion into three classes:

- **global**—unrestricted geo-localization at the planet scale (Fig. 2.1(a)),
- **city-scale**—geo-localization in urban environments (Fig. 2.1(b)),
- **natural**—geo-localization in natural (non-urban) environments—*e.g.*, in the mountains (Fig. 2.1(c)).

The goal of global methods is to localize query image without a prior assumption about the environment type. Localizing a single image in the whole world is an appealing idea, but the existing methods provide low accuracy. Existing works consider the localization successful if the method localizes the query image within 200 km from the ground truth position [77].

City-scale methods deliver higher localization accuracy, assuming the query image resides in a specific urban area. Natural methods are specialized as well—the published methods target specific kinds of natural environments, such as deserts or mountains. There are principal differences between urban and natural environments that determine the complexity of the respective geo-localization problem:

Data Availability. Dozens of photos of attractive places and landmarks in highly populated areas—Flickr API returns more than 200,000 photographs containing the tag "Eiffel Tower" and more than 100,000 photographs containing the tag "Statue of Liberty" (2016). Such an abundance of data enables an image to image search with Bag of Words (BOW), feature-based techniques, and SfM model matching.

Well Defined Objects. Human-made objects with distinctive and stable appearance, such as buildings, bridges, or road signs, can be well recognized and matched. Moreover, such objects' mutual arrangement in space is often unique, which we can use for localization. On the other hand, in natural environments, objects are relatively difficult to match-—*e.g.*, mountains, foliage, and clouds. Those are difficult to recognize because of inconsistent appearance (weather and illumination changes, vegetation growth) and frequent occlusion of such objects in the real world.

Repetitive and Self-Similar Patterns. Urban environments contain repetitive objects like windows, lamps, and logos. In natural environments, we can see a lot of fractal and self-similar patterns. All these aspects make the visual geo-localization a problematic task.

Such specific issues narrow down the options for solutions of geo-localization in a particular environment. This chapter presents a broad overview of visual geo-localization methods in connection with the introduced classification. We summarize all reviewed localization methods in Tab. 2.1.

method	class	environ.	test area	localiation success	max. err.
Robertson 04 [152]	image-based, retrieval	city	single street	95%	N/A
Zhang 06 [228]		city	city part	72% on ICCV 2005 Cont.	16 m
Schindler 07 [166]		city	single city	70%	10 m
Zamir 10 [225]		city	240 km of street-view	78%, vs. 39% [166]	250 m
Chen 11 [34]		city	single city	65%	N/A
Johns 11 [87]		city	landmark	N/A	N/A
Zamir 14 [226]		city	several cities	N/A	N/A
Zamir 14a [224]		city	several cities	44%	100 m
Arandjelovic 15 [7]		city	Pittsburgh250k	86.0% top-1 candidates	25 m
Liu 19 [119]		city	Pittsburgh250k	89.0% top-1 candidates	25 m
Ge 20 [58]		city	Pittsburgh250k	90.7% top-1 candidates	25 m
Hays 08 [77]		global	Earth	16% on IM2GPS [77]	200 km
Zheng 09 [230]		global	Earth landmarks	accuracy 80.8%	N/A
Hays 15 [78]		global	Earth	32.1% on IM2GPS [77]	200 km
Vo 17 [207]		global	Earth	47.7% on IM2GPS [77]	200 km
Li 09 [112]		global	Earth landmarks	40.58% visual&tags	1 of 500 landm.
Mishkin 15 [127]		global	place	P/R: $\frac{0.821}{0.825}$	1 frame
Irschara 09 [83]	image-based, SfM	city	landmark	39% within top-10 candidates	N/A
Li 10 [114]		city	several cities	92.4% (Rome)	400 m
Sattler 11 [158]		city	several cities	97.6% (Rome)	400 m
Sattler 12 [159]		city	several cities	99.1% (Rome)	400 m
Sattler 12a [161]		city	Aachen [161], Vienna [83]	74-83%	N/A
Li 12 [113]		city	1 K of landm.	73% on Quad [42] 90%, images under 10 m	N/A
Hao 12 [70]		city	landmark	N/A	N/A
Bergamo 13 [23]		city	landmark	95% on Lan.-3D [70] 63% on Lan.-620 [23]	N/A
Svärm 14 [188]		city	Dubrovnik [114]	0.9975% (Dubrovnik)	400 m
Sattler 15[157]		city	San Fr. [113] Landmarks	62.5% (San Fr.)	N/A
Zeisl 15 [227]		city	San Fr. [34], Dubrovnik [114]	0.9975% (Dubrovnik)	400 m
Kendall 15 [95]	class., regress.	city, indoor	Cambridge Landmarks	2 m, 3° outd. 0.5 m, 5° ind.	N/A
Brachmann 17 [26]		city, indoor	7 scenes [177] indoor	62.5%	5 cm; and 5°
Brachmann 18 [25]		city, indoor	7 scenes [177] Cambridge Landmarks	76.1%	5 cm; and 5°
Weyand 16 [213]		global	Earth	37.6% on IM2GPS [77]	200 km
Seo 18 [173]		global	Earth	46.4% on IM2GPS [77]	200 km
Müller-Budack 18 [131]		global	Earth	51.9% on IM2GPS [77]	200 km
Izbicki 20 [84]		global	Earth	39.4% on IM2GPS [77]	200 km
Talluri 92 [190]	mul. DEM	natural	148 km^2	N/A	N/A
Stein 95 [186]		natural	298 km^2	N/A	N/A
Naval 97 [146]		natural	N/A	N/A	N/A

continued on the next page...

. . . continued from the previous page

method	class	environ.	test area	localiation success	max. err.
Naval 98 [133]	multi. DEM	natural	900 km^2	avg. err. 393 m	N/A
Woo 07 [215]		aerial, natur.	2.28 km^2	N/A	N/A
Baatz 10 [13]		city	single city	35%, or 85%	N/A
Ramalingam 10 [150]		city	single city	avg. err. 2.8 m	N/A
Baatz 12 [14]		natural	40 000 km^2	88%	1 km
Tzeng 13 [200]		natural	10 000 km^2	N/A	N/A
Baboud 11 [15]		natural	28 photos in the Alps, Rocky Mnts.	86%	<0.2°
Porzi 14 [143]		natural	100 places in the Alps	avg. err. 1.87°	5.22°
Hammoud 13 [69]		natural	20 000 km^2	49%	14 km
Chen 15 [37]		natural	10 000 km^2 (America, Asia)	60%	4.5 km
Hakeem 06 [67]	multi. SLAM	city	campus	avg. err 6 m ICCV Cont. 2005	N/A
Conte 09 [40]		natural	N/A (S. Sweden)	N/A	N/A
Larnaout 12 [108]		city	city-center	N/A	N/A
Larnaout 13 [109]		rural, city	rural, city	N/A	N/A
Middelberg 14 [125]		city	40 km^2	<1 m	N/A
Jacobs 07 [85]	multi. other	global	Pennsylv., Maryland	avg. err. 71.8 km	N/A
Gallagher 09 [57]		global	Earth	33% on IM2GPS[77]	200 km
Kalogerakis 09 [89]		global	Earth	58% on IM2GPS [77]	400 km
Baatz 10 [13]		city	single city	Earthmine 85% Navteq 35%	N/A
Kelm 11 [94]		global	Earth	10%	1 km
Kelm 11a [93]		global	Earth	35%	1 km
Lin 13 [116]		global	1 600 km^2	17.37%	N/A
Aubry 14 [10]		city	landmark	55% good matches	18% no match
Viswanathan 14 [206]		aerial, natur.	approx. 0.1 km^2	31% matches for top 10% cand.	N/A
Ardeshir 14 [8]		city	10 km^2 Washington DC	60% for top 20% cand.	N/A
Lin 15 [117]		city	several cities	80%	20% of cand.
Workman 15 [216]		global	40 000km^2	22.7%	N/A

Table 2.1: Overview and properties of geo-localization methods. Test area defines the area on which the method has been tested in the original publication, localization success (local. succ.) denotes the best result achieved with given method, and maximum error denotes the maximum distance from the ground truth position which is considered to be correct localization. **Abbreviations**: *multi.* = methods using data of multiple modalities, *cont.* = contest, *cand.* = candidates, *P/R* = precision/recall, *landm.* = landmarks, *mnts.* = mountains, *tags* = method uses also user defined tags for localization, *San Fr.* = San Fransisco.

2.2 Image-based Methods

Image-based methods are used when sufficient amount of reference images is available. *Image retrieval* uses big databases of GPS-tagged images to infer a query image's location by retrieving similar images using various matching algorithms. *Localization by regression* uses machine learning to learn a model to directly predict the camera parameters (position and/or orientation). Alternatively, *localization* approaches using *classification* discretize the space of camera parameters into a set of disjoint classes and learn a model to predict a class given an input image. *Structure from Motion* (SfM) localization methods use a 3D reference model constructed using geometrical relationships between many overlapping images. Thanks to this fact, not all images need to contain explicit GPS tags.

2.2.1 Image Retrieval

Image retrieval is a set of methods to search for similar images in an extensive image database. Usually, the query image location is inferred based on the location of the most similar database images [152, 228, 166, 87, 225, 226, 224, 127].

Robertson and Cipolla [152] published one of the first attempts to image localization by image retrieval. They created a database of two hundred photographs of rectified facades in Cambridge city center. For rectification, they used vanishing points estimation by Kosecka and Zhang [100]. They manually annotated facade positions using the 2D map to connect each facade with meaningful coordinates. For matching, the authors used the sum of squared differences between patches centered around Harris key-points. The method does not scale well, since it matches the query image against all the database images, which would lead to prohibitive run times on more extensive image databases.

Zhang and Kosecka [228] extended the former approach using a database of SIFT feature descriptors [122] detected on GPS-tagged images. The authors implemented the coarse matching stage as simple voting to every document in the database causing high computational complexity. Their method verifies the best five candidates and sorts them using RANSAC [55]. The method finds the final location by triangulating the best candidates.

Schindler *et al.* [166] developed another city-scale image localization method based on image retrieval. Their publicly available test dataset includes 20 km of street-side imagery.

Johns and Yang [87] studied the problem of place recognition. They improved the BOW technique [180] by clustering the image database of 200,000 images to visually similar scene models (landmarks). However, their results show only marginal improvement compared to the standard BOW technique.

Zamir and Shah [225] used a dataset of 100,000 geo-tagged images downloaded from Google Street View. They used a nearest-neighbor tree search with additional steps of pruning and smoothing for better accuracy. Furthermore, they developed a *Confidence of Localization* measure, which quantifies the reliability of the localization of a particular query image using Kurtosis of a normalized voting space.

Zamir *et al.* [226] proposed a robust method operating on an image database with noisy GPS tags. For each query image, the method finds several matches from the image database to form triplets. With the assumption of noise-free GPS tags, the method can estimate the query image's geo-location directly from the triplet. To allow robust estimations under noisy data, the authors propose to use random walks.

Zamir and Shah [224] aimed to further improve nearest-neighbor matching by pruning outliers and incorporating approximate feature matching using a Generalized Minimum Clique Problem (GMCP). The authors compare their method to Schindler *et al.* [166] and their previous work [225]. They show that the new method has lower localization error; it was able to localize more than 55% of the query images within the error of 250 m, whereas their previous method [225] localized 50% and Schindler *et al.* [166] localized only 46% within the same error.

Arandjelovic *et al.* [7] proposed a novel neural network layer called NetVLAD to produce a global feature descriptor for a given image. They casted the city-scale place recognition problem as image retrieval by searching nearest neigbors in the database of global descriptors. The NetVLAD correctly localized around 86% of top-1 candidate images of the Pittsburgh250k dataset.

More recent approach by Liu *et al.* [119] builds on the NetVLAD and proposes a novel Stochastic Attraction and Repulsion Embedding loss function (SARE). The SARE loss function aims to minimize similarities among inter-place images while maximizing similarities among intra-place images. This approach correctly localized almost 89% of top-1 candicates of the Pittsburgh250k dataset.

A self-supervised approach for learning global descriptors was proposed by Ge *et al.* [58]. The authors propose a method to gradually increase the difficulty of the positive and negative training samples by self-supervising the image-to-region similarities. This method achieves state-of-the-art accuracy on both Pittsburgh250k and Tokyo 24/7 datasets. On the Pittsburgh250k this method localized 90.7% of top-1 candidates.

An interesting problem of place recognition in changing conditions, such as changes between day and night or winter and summer, was explored by Mishkin *et al.* [127]. They adopted a BOW method with multiple detectors, descriptors, view synthesis, and adaptive thresholding to cope with extensive visual changes in the environment.

Hays and Efros [77, 78] published the first global visual geo-localization method. They created a database of various features from 6 million images distributed around the whole Earth. To estimate a query image's location, the authors used the retrieved nearest neighbors' density using various handcrafted features. At the threshold of 200 km from the ground truth the former approach [77] localized 15% of images, and the latter approach [78] localized 32.1% of images on the IM2GPS test set.

The follow-up work by Vo *et al.* [207] revisited the IM2GPS approach by using features from deep neural networks for image retrieval at the global scale. Similarly to PlaNet [213], they trained a CNN for classification, however, at the inference they used the network activations as features for image retrieval, instead of classification. They show their approach needs lower amount of training data and is able to achieve more accurate results compared to PlaNet [213]. Their approach localized 14.4% of images within the 1 km and 47.7% of images within the 200 km distance from the ground truth.

Since landmark recognition techniques lie at the border of our interest, let us review them only briefly. Li *et al.* [112] use the BOW technique combined with multiclass Support Vector Machine (SVM) to learn landmark classification. Similarly, Avrithis *et al.* [11] used an improved BOW method to study the problem of separating landmark and non-landmark images. Zheng *et al.* [230] combine GPS-tagged images from online services and a textual tour guide with unsupervised learning to build a world-scale landmark database. Chen *et al.* [34] studied landmark detection on mobile devices using on-board GPS estimates. The authors also published a dataset for landmark recognition and localization.

2.2.2 Localization by Regression and Classification

Recently, researchers proposed deep learning to directly predict the camera location from scene observations using a forward pass through a CNN [95, 213, 26, 25]. Once trained, the regressor function can estimate the camera parameters directly given the query image. This approach's advantage resides in learning a relatively compact representation of the whole scene, unlike the image retrieval or SfM methods, which need to operate on a database of real images and their descriptors with a large memory footprint. On the other hand, for each scene, a specific regressor is needed, and training the regressor could be quite expensive.

Kendall *et al.* [95] used an SfM model to train a convolutional neural network called PoseNet for camera localization. Their experiments operate on 50 000 m^2, with the reported errors 2 m and 3° in outdoor areas, and 0.5 m and 5° in indoor areas.

DSAC and DSAC++ [26, 25] use a neural network to predict each image's 3D representation, so-called scene coordinates. The Perspective-*n*-Point (PnP) algorithm in a RANSAC loop [55] uses the estimated 3D coordinates to estimate the camera's absolute pose. DSAC++ can achieve accurate results on a known scene—translation error is in millimeters, and rotation error is in the order of tenths of a degree.

PlaNet [213] aims to localize images across the whole Earth (at the global scale) by clustering the whole Earth into a large number of cells of variable size. The authors trained a neural network to classify a query image into the database of cells using 126 million training images. Although this approach works at the global scale around the whole Earth, it is relatively inaccurate—on the IM2GPS test set it was able to localize 8.4% of images within 1 km, and 37.6% of images within 200 km from the ground truth.

A follow-up work, CPlaNet [173] introduces a method to tackle the problem of the trade-off between cell size and number of training images within a cell. The basic idea is to partition a world into several coarse partitionings; for each partitioning a separate classifier is trained. The paper proposes a novel combinatorial partitioning method, which merges separate classifier outputs into a fine-grained classification. Especially for fine-grained scales its accuracy is almost doubled compared to the original PlaNet [213]: on the IM2GPS test set it localized 16.5% of images within 1 km and 46.4% of images within 200 km.

A paper aimed at global geo-localization exploiting the hierarchical nature of the classification in combination with photo's scene content (*i.e.*, indoor, natural, or urban) was introduced by Müller-Budack *et al*. [131]. The performance of this method is better compared to CPlaNet [173], and it needs just 4.7 million images for training, while CPlaNet consumed 30.3 million of images. This approach currently delivers state-of-the-art results in global geo-localization: on the IM2GPS test set it localized 16.9% of images within 1 km and 51.9% of images within 200 km.

Izbicki *et al*. [84] approached the global geo-localization by introducing the Mixture of von-Mises Fisher loss function, which is similar to a Gaussian Mixture Model adapted to a Spherical surface. This approach could be used in a hybrid mode, which combines classification and retrieval approaches together. This approach is able to deliver significantly better results on coarser scales compared to CPlaNet [173] and the revisited version of IM2GPS [207].

A recent study [162] suggests that direct regression of camera parameters, such as PoseNet [95] or PlaNet [213], behaves and performs similar to image retrieval, which is less accurate than methods leveraging the 3D structure. On the other hand, DSAC++ [25] operates by scene coordinate regression and performs precisely on a small scale, but it seems it cannot generalize to large-scale scenes [162].

2.2.3 Structure from Motion

Structure from motion (SfM) is a set of reconstruction methods and strategies for computing a 3D scene from an unordered set of photographs. The core of SfM involves computing a relative pose between two cameras observing the same scene, by solving the 5-point problem [134]. Additional photographs can be added to the reconstruction by computing absolute pose between camera images and a known set of 3D points, typically by solving the Perspective-n-Point (PnP) [55] algorithm. These algorithms are usually used iteratively to incrementally add new photographs to the reconstruction, further refined by non-linear optimization of the camera parameters and 3D points called global Bundle Adjustment [199].

Tens of million images available online can be used to create large SfM [72, 27, 184, 185, 1, 65, 42, 80] models. For instance, Heinly *et al.* [80] automatically created models of many places worldwide from 100 million photos from the Yahoo image dataset [191] in six days on a single computer.

SfM models are usable in highly urbanized areas and near dominant landmarks. Irschara *et al.* [83] used several hundreds of photographs to create SfM models of Vienna's most famous landmarks. They search for relevant photographs in the SfM model by the standard image retrieval (BOW) approach. They successfully registered the majority of frames of four test videos and test images. The authors also presented a compression technique to reduce the number of images needed to cover the 3D scene.

Li *et al.* [114] developed a location recognition approach that prioritizes features from an SfM model and matches them against query features. They show that defining priorities based on properties of features in the SfM model and application of Feature-to-Point (2D-to-3D) matching play a vital role in the improvement of matching performance.

Sattler *et al.* [158] propose a technique of direct 2D-to-3D matching. They assign a feature descriptor to each visual word and match query feature descriptors directly to descriptors in relevant visual words. They show an improvement in matching performance while keeping reasonable response times (fractions of a second). In follow-up work by Sattler *et al.* [159], the ideas of 2D-to-3D and 3D-to-2D were combined and formulated into an *Active Correspondence Search*, which improved both the time and the matching performance.

Furthermore, Sattler *et al.* [161] studied problems of image retrieval methods connected to localization. Algorithms using direct feature descriptor matching outperform classical image retrieval approach by 15%. The authors identified and addressed the image retrieval approach's performance problems by introducing selective voting. This method slightly outperformed the direct descriptor matching.

Among the first works addressing large-scale localization based on an SfM model was Li *et al*. [113]. They presented a method able to cope with hundreds of thousands of images using *a co-occurence prior for RANSAC* and a *bidirectional matching of image features with 3D points*, which is a similar idea to the *Active Correspondence Search* presented by Sattler *et al*. [159].

Bergamo *et al*. [23] used an SfM model to learn a random forest codebook for Landmark classification. Other authors also approached the problem of landmark classification [148, 70], but it is out of this survey's scope.

Swärm *et al*. [188] incorporated the knowledge about gravity direction in the query image obtained from gravitational sensors. Their method can handle a large amount (up to 99%) of outliers.

Localization on large datasets (hundreds of thousands of images in the SfM model) poses new problems, namely a large memory footprint of the model and the Scale Invariant Feature Transform (SIFT) descriptor ratio test's strictness. These problems are approached by Sattler *et al*. [157] by quantizing descriptors to reduce the search space while incorporating a new voting strategy to remove ambiguous matches.

The work by Zeisl *et al*. [227] on large scale geo-localization using the SfM model also tackles the problem of a large fraction of outlier matches. The authors build on Svärm *et al*. [188], utilizing geometric constraint of gravity direction on camera and incorporate them with additional constraints into the camera pose voting.

2.3 Methods Using Data of Multiple Modalities

Unlike image-based methods, methods leveraging multiple modalities use additional input data to find camera location for a query image. A popular choice is a cross-domain matching of a query image and a *terrain model*, with the utilization of features like horizon lines, ridges, and edge maps. *Simultaneous Localization and Mapping* aims to create a map of an unknown environment and simultaneously localize a camera in that environment. Researchers also proposed *methods using other input data* like orthophoto maps combined with attribute maps, bird's eye, or satellite weather imagery. In this domain, we consider outdoor, non-urban environments due to areas with a lower population density.

2.3.1 Methods Using Terrain Models

The first visual geo-localization works' primary motivation was mobile robots' and planetary rovers localization in outdoor environments. Talluri and Aggarwal presented an early work [190, 189] covering this topic. They use a DEM model and a robot equipped with a digital compass, an altimeter, and a monocular camera, that can be panned and tilted. They propose to localize by matching horizon lines extracted from a query image with those rendered from DEM. The authors conducted experiments on 1.41 km^2, with the area sampled uniformly with the distance of samples of 30 m.

Stein and Medioni [186] use horizon lines for localization as well. They create a database of synthetically rendered 360° horizon lines using a DEM. They approximate horizon lines by polygons, from which they create a database. They extract the horizon from an input query image semi-automatically and encode it into the same format as horizons in the database. Finally, they match the query horizon line with the database and geometrically verify the best candidates.

Naval *et al*. [146, 133] further studied the localization using horizon lines. In these works, they extract the skyline from a query image by a multilayer perceptron. They use the peaks as local feature points, which they detect in both the query image and the DEM. The authors calculate the query image's pose using three feature points from the database via minimization of error function using non-linear least squares.

Woo *et al*. [215] studied unmanned areal vehicle (UAV) navigation in mountain areas using DEM and infrared (IR) images with known altitude using an altimeter. They used the infrared spectrum to tackle the visibility problems during the night and bad weather conditions. The authors extracted the peaks from a series of frames and used a factorization method to reconstruct a spatial configuration of peaks in 3D. Next, the authors proposed matching the 3D positions of peaks extracted from the query image to peaks extracted from DEM, and hypothesizing the pose. Finally, they rendered an artificial horizon from the DEM at the hypothesized location and aligned it with the query horizon line to confirm or reject the estimated location.

Ramalingam *et al*. [150] published a city-scale visual geo-localization method based on fisheye images of the urban canyons. The method takes an omni-skyline image, extracts the skyline defined by buildings, and matches this skyline with a database of synthetically rendered skylines. The method is usable in cities which have very tall buildings, like New York.

Produit *et al*. [145] developed a method to estimate a full camera pose from point correspondences between the rendered DEM covered with an orthophoto texture. For matching pixel patches located at the corners of salient edges they used a cross-correlation. They estimate a full camera pose using the matches filtered with Random Sample Consensus (RANSAC).

Hammoud *et al*. [69] extend the extracted horizon line from a query image by LIDAR and Hyper-Spectral Land Use/Cover imagery. They match the inputs separately and combine them by linear fusion into a single probability map. The authors validated their approach on 100 query images on two world regions, having 10,000 km^2 each.

Baatz *et al*. [14, 163] were the first to develop factually large-scale visual localization solution outdoors (on an area of 40,000 km^2). The method uses an extensive database of extracted features from horizon lines, called contourlets. The contourlets are dense representations of normalized and smoothed horizon lines stored as a single integer. For localization, they use a BOW approach to retrieve the best 1000 candidates, which are geometrically verified to find the best matching locations. Thanks to direction & location voting strategy and geometrical verification of horizon lines, the method can estimate both camera's location and coarse heading.

Tzeng *et al*. [200] presented a similar work to Baatz *et al*. [14]. The idea of using a database of horizon features generated from a rendered DEM and searching for horizon features from a query image is the same. The difference is that they use concavities of horizon line parts as local features.

Chen *et al*. [37] presented an advanced approach based on horizon lines. The authors build on the approach presented by Saurer *et al*. [163], and they extend the local feature descriptor utilizing multiple ridgelines, not only the horizon line. The feature extraction is the same as in Saurer *et al*. [163]. The crucial difference is in the voting stage of BOW, where the documents are voting not only for the horizontal but also for the vertical direction. The authors tested their method on 10,000 km^2 and showed that their results were better than in Saurer *et al*. [163].

2.3.2 Simultaneous Localization and Mapping

Visual Simultaneous Localization and Mapping (V-SLAM) is also relevant to the topic of visual geo-localization when performed outdoors. Generally, SLAM methods make use of various inputs, like RGB image combined with depth, stereo, lidar sensors, or GPS. We focus on the works relevant to visual geo-localization, surveying the works utilizing only the single-camera input. Since SLAM methods focus on continuous localization in time, we separated these methods from visual geo-localization of a single image.

An approach by Middelberg *et al*. [125] for six-degrees-of-freedom (6-DOF) localization on mobile devices uses a large offline SfM point cloud at the server and a small keyframe-based SLAM [97] model on the mobile device. The keyframes are matched with the offline SfM model to avoid drift, while intermediate frames are processed on the device to estimate the motion frame-by-frame.

Hakeem *et al*. [67] proposed an offline method for estimating a moving camera trajectory. They match keyframes with a GPS-tagged photographs database, and from the best matches they calculate essential and fundamental matrices to recover the camera pose. They use a triangulation step to disambiguate the scale. Next, they interpolate the obtained locations using B-splines to obtain a smooth trajectory.

Conte and Doherty [40] used a GPS-tagged image database in combination with a KLT feature tracker [194] to address the problem of GPS signal outages of an UAV. The visually tracked position was fused with the inertial measurement via on-board sensors through a Bayesian framework.

Vaca-Castano *et al.* [202] built a method on top of the localization method by Zamir and Shah [225] for trajectory estimation in a city. Each keyframe is localized using the discussed method, and Bayesian filtering enforcing temporal coherency is used. As the results are often noisy and exhibit false loops, they construct the final trajectory using a minimum spanning tree (MST) algorithm.

Larnaout *et al.* [108, 109] combine classical SLAM methods with an elevation constraint taken from a DEM, because SLAM vehicle's height is constant. They also add 3D building models as a constraint to the reconstructed 3D point cloud.

2.3.3 Methods Using Other Input Data

Baatz *et al.* [13] researched a method for localization in the urban environment. They use panoramic street-view images and extruded floorplans of buildings to build a rectified image database (mapping the facades onto the extruded 3D models). A query image is also rectified based on vanishing points, which reduces the matching problem to the 2D homothety.

Data-driven solutions aim to learn the relationship between a photograph and the land cover appearance based on a geo-tagged ground-truth dataset. Lin *et al.* [116] create a geo-database from several corresponding data sources and match an input query photograph with the triplets of ground-level images, an aerial orthophoto map, and an attribute map.

The idea of cross-view matching was researched by Workman *et al.* [216], who approached the problem by adapting a CNN (pre-trained on Places [232] dataset) to extract similar features from ground-level photographs and aerial orthophoto maps. They used nearest neighbors as candidates ranked by calculating the euclidean distance between the ground level and aerial features. The authors also developed a large dataset containing over 1.5 million geo-tagged matching pairs. The authors claim their method is state-of-the-art in cross-view geo-localization, supported by a 6% improvement compared to the previous work by Lin *et al.* [116].

Lin *et al.* [117] presented similar work to Workman *et al.* [216]. They use a CNN for the cross-view matching, but they use Google Street View and aerial "bird's eye" imagery, which is captured tilted compared to classical aerial orthophoto imagery taken orthogonally to the terrain. They used a CNN pre-trained on ImageNet and Places [232] databases. The results are currently far from the practical application; 20% of top candidates contain 80% of correctly localized queries.

Castaldo *et al.* [32] approached the cross-view matching problem from a different perspective. From a query image they extract a semantic segmentation and generate a rectified top-down view using vanishing line of the ground plane. From both rectified query semantic segmentation and the GIS map a local descriptors encoding the layout of semantic regions is computed. The location is retrieved by matching the query descriptors with the descriptors extracted from the map. The authors experimented on the area of 159 km^2 of the District of Columbia, USA. According to the experiments, 20% of top condidates contain roughly 77% percent of correctly localized queries.

Aubry *et al.* [10] developed a method to register an artistic painting with a 3D model, which also implies the camera's pose. For matching, they mention the possibility of using exemplar-based SVM classification introduced by Shrivastava [178]. Based on this approach, they developed a new method to avoid training SVM classifiers. They tested the method on various historical paintings, which they successfully registered with the 3D model.

Viswanathan *et al.* [206] developed a robot localization method by matching Google Street View panorama to an aerial orthophoto map. They warp the street view panorama to the bird's eye view (top-down) and use standard matching techniques using various features like SIFT, Speeded-Up Robust Features (SURF), and Fast Retina Keypoint (FREAK). In their scenario, SIFT proved to have a stable performance throughout the test set.

Ardeshir *et al.* [8] exploit semantic information from a geographic information system (GIS) database, such as locations of fire hydrants, traffic signals, road signs, and other objects to improve object detection. Image metadata as GPS location, FOV, and heading are used as a hypothesis to match the objects in the query image against the objects obtained from the GIS database under a given viewpoint. Based on object detection, the authors also developed a method of camera localization. The method generates location hypotheses on a uniformly sampled grid, excluding the areas covered by buildings. For each hypothesis, the method detects the objects and calculates a location-orientation score.

Senlet *et al.* [172] proposed an approach for localization of aerial query images. They use semantic segmentation to detect buildings in orthophoto satellite images. The authors use the Geometrical Hashing of the spatial relations between the segmented buildings to create a geospatial database. Hashes computed from building segmentation of the aerial orthophoto query image are used to search the database. The authors conduct their experiments on an area of 16 km^2 of a city map densely covered with more than 7,000 buildings.

Armagan *et al.* [9] researched an iterative method to fine-tune camera pose based on alignment of semantic areas of buildings. They use untextured 2.5D model of buildings to render a synthetic image from an initial camera pose estimate. To detect semantic segmentation related to the *building* class in the query image, they use Fully Convolutional

Networks (FCN) [120]. They trained a CNN to predict a best direction to improve camera pose based on the query image segmentation and the rendered 2.5D buildings' model. The authors use the estimated direction to improve the camera pose, render a novel view and iterate until convergence. The experiments illustrate that the method is consistently able to improve the initial imprecise camera pose, which can be off by 25 meters and 50 degrees.

The geographic coherence in image sequences may also be used for camera localization. Jacobs *et al.* [85] exploit sequences of frames from static outdoor cameras correlated with satellite imagery for location estimation. Kalogerakis *et al.* [89] learn human travel priors from a 6 million database of images from the Flickr web service. Their approach can geo-localize image sequences from the user gallery, using timestamps to calculate probable locations based on the learned prior. Kelm *et al.* [94, 93] use video keyframes combined with textual features to find the most probable regions of origin.

Multimodal approaches exploiting textual tags or other information exist as well. The global geo-localization method by Gallagher *et al.* [57] used a database containing over million of geo-tagged images and user-defined textual tags. They used user tags from a query image in the matching process in parallel with several other visual features like GIST, color histograms, tiny images, and a bag of textons.

2.4 Camera Orientation Estimation

Camera orientation estimation problem is also related to visual geo-localization. Some visual geo-localization methods are designed to retrieve the camera orientation, especially SfM [83, 113, 188, 125, 95], or horizon-based and DEM matching approaches [145, 69, 200, 37, 163]. However, some methods, like image-based and cross-view visual geo-localization methods, estimate 3-degrees of freedom (DOF) position only and cannot deliver camera orientation [77, 116, 117, 213]. In such cases, the geo-localization and camera orientation methods could be used together in order to retrieve full 6-DOF pose.

Kosecka and Zhang [100] presented algorithm for camera orientation estimation based on vanishing points. This method is suitable for urban indoor and outdoor scenes, as the detection of vanishing points is based on line segments. The line segments can be detected in urban scenes easily, while in natural scenes they are present sparsely.

Several approaches for camera orientation estimation for natural scenes exist. Behringer [22] matches synthetic panoramic horizon line to horizon line detected in query image. This approach was extended by Baboud *et al.* [15], who presented an algorithm for robust silhouette matching. Since it matches the synthetic and the query edge maps, it is much more robust to occlusion than methods using horizon line only. Baatz *et al.* [14] published camera orientation algorithm based on matching sematnic areas in the image, like forests or rivers. Efficient camera orientation refinement was approached by Porzi *et al.* [142]. They use smartphone sensors as an initial estimate, which is refined by silhouette matching algorithm similar to [15].

2.5 Applications of Visual Geo-localization

Several of works suggest many exciting applications of visual geo-localization. Let us briefly review the most intriguing applications related to our work. Visual geo-localization is a high-level problem and it is itself an application of many computer vision algorithms; from a query image or a video, we obtain a geographic location. However, this information can be further processed and used.

In online applications, people can try their visual geo-localization abilities. *GeoGuessr*[4] site uses Google Street View panoramas as query images, and people are supposed to guess the location. *View From Your Window Contest*[5] is a similar website, where challenging sets of images are to be geo-localized by people. Weyand *et al.* [213] has recently published an evaluation of their geo-localization system, which was able to beat geo-localization estimations made by people systematically.

Various methods for digital photo enhancement were presented in Deep Photo [99]. The knowledge of location and orientation is crucial for methods like model-based haze removal. We can also attain other tricks — as Kopf *et al.* [99] showed, we can alter illumination in the original image with the synthetic one, and we can also augment the image by labels or artificial segments like paths or motorways. With a collection of precisely aligned photographs, we may also leverage photo un-cropping methods to create a photograph with a larger field of view [175], or to faithfully complete missing regions of a photograph [234]

Kendall *et al.* [95] published a nice demo of their relocalization framework[6]. This online application can estimate the exact pose of the query image in the trained area. With such an application, people can localize themselves using their smartphones even without GPS.

Like Junior [128] or UAV's, autonomous vehicles are indeed another application of visual geo-localization. Such devices use several inputs, like LIDAR, GPS, video, and more to preserve location recognition's robustness. The vehicles need to solve many problems aimed by state-of-the-art in computer vision, like pedestrian and traffic sign detection, self-localization, localization of other cars in traffic, reference speed measuring, and more.

[4] `https://www.geoguessr.com/`

[5] `http://dish.andrewsullivan.com/vfyw-contest/`

[6] `http://mi.eng.cam.ac.uk/projects/relocalisation`

Google Lens (formally Goggles) is a mobile application from Google. It can recognize objects and identify landmarks, as pointed out by Chen *et al.* [34].

2.5.1 Photography Presentation

Knowing a photograph position can also be utilized by methods aiming at photography presentation. Previous research has found that we can facilitate users' spatial understanding by incorporating animation [21], spatial context [212, 211], interaction [102], and panoramas [43].

Chippendale *et al.* [38] summarized possible future geo-localized photography applications like automatic creation of PhotoOverlays in Google Earth, or photographs augmented with peak names and GPS tracks. Snavely *et al.* explored presenting photographs in a 3D environment in their PhotoTourism paper [184], which uses SfM to reconstruct 3D point clouds for famous landmarks. They also designed automatic path planning and photo exploration in the reconstructed environment [183]. Subsequent work uses similar techniques for automatic path planning [104] and effective photo acquisition of a site of interest [182]. Hyper-lapse videos [98, 209] yield a similar visual experience by smoothing and stitching egocentric videos.

Exploring spatially positioned photographs without 3D reconstruction has been proposed as well. Kaneva *et al.* [90] use image retrieval to find similar images, stitch them together, and create a fictitious photorealistic virtual space. Tompkin *et al.* [195] combine videos with a panoramic image, so the user better understands the mutual orientation and temporal relationship of different videos taken from the roughly same place. Veas *et al.* [203] studied spatial understanding and navigation in outdoor environments using video streams from several cameras. Video presentation in a 3D space has also been used to improve medical responses [124].

2.6 Chapter Summary

We classified the surveyed methods according to the type of input data, into *image-based methods* and *methods using data of multiple modalities*. Moreover, we introduced a second categorization based on environment, for which the geo-localization method was designed: *global*, *city-scale*, and *natural*. The *image-based methods* were used mainly for urban areas, while *methods utilizing multiple modalities* were used mainly for localization outside city borders—in natural environments.

For *image-based methods*, we identified three main geo-localization approaches. The first is image geo-localization using *image retrieval*, which retrieves the most similar image to the query image from a geo-tagged database. The second approach uses the 3D SfM model of the scene or a geo-tagged database of images to train a *classifier or regressor*, which estimates the query image's location directly. Finally, the third is geo-localization by querying a 3D model created from many overlapping images using *structure from motion (SfM)* and calculating a camera pose relatively to the 3D model. Please note that since this approach leverages geometrical correspondences between the 3D scene images, we don't necessarily need to know the real-world position for all the images contained in the 3D model.

For *methods using data of multiple modalities*, the approach of horizon line matching is a widespread technique [190, 189, 186, 146, 133, 14, 163, 37]. Another popular technique is a *cross-view* matching approach introduced by Lin *et al.* [116] and further studied in other variants [216, 117, 32].

While image-based solutions are well established and achieve precise results, their use is limited. Researchers proposed algorithms for fast and precise geo-localization for urbanized areas without the need for a GPS sensor [125]. When we travel outside the borders of a city, the situation is different. In natural environments, we still lack fast and, more importantly, precise algorithms. Still, the researchers need to spend much work to get similar precision as in the urbanized areas. For example, Saurer *et al.* [163] consider the query image as correctly localized if the found location is up to 1 km from the ground truth. This is is still far from the results obtained by Middelberg *et al.* [125], who report the localization error in meters. In the case of horizon-based localization proposed by Saurer *et al.* [163], 40% of query images need user interaction to discover the horizon line, mainly due to tree occlusions that arise in real-world photos quite often. Furthermore, it is difficult to address the horizon occlusion by fog or clouds in this scenario. We need more robust features such as edges or semantic segments, like areas of forests, glaciers, or bodies of water for such situations.

Finally, we showed popular applications of visual geo-localization methods. The applications range from autonomous vehicles [128] through photography enhancement [99] to photography presentation [184, 183] and visualization [38].

Chapter 3

Evaluation Practices in Visual Geo-localization

Existing and coming methods solving visual geo-localization tasks need appropriate evaluation methods to illustrate their strengths and weaknesses. Datasets for these tasks are nowadays used not only for evaluation purposes—with the advent of parametric trainable methods (usually neural networks)—they are also used to optimize model parameters using the training set. In this chapter, we review existing datasets relevant to our work (Sec. 3.1). Next, we review evaluation metrics for visual geo-localization and introduce evaluation metrics we use throughout this thesis (Sec. 3.2).

3.1 Datasets for Visual Geo-localization

In the following text, we review existing datasets for visual geo-localization and categorize them according to their content. Image-based datasets presented in Sec. 3.1.1 contain images coupled with their position. Geometry-based datasets (Sec. 3.1.2) contain also geometric relationships between images and are usually acquired using photogrammetric methods, such as SfM, or with the help of laser scanning methods, such as Light Detection And Ranging (LIDAR). Visual geo-localization datasets of images with multiple modalities (Sec. 3.1.3) contain also additional metadata, such as semantic segmentation, depth, silhouettes, etc. The datasets mentioned in this survey are summarized in Tab. 3.1.

3.1.1 Image-based Datasets

Datasets of this kind are relatively easy to collect—researchers usually use ground-level imagery with GPS tags downloaded from various internet services, like Google Street View or Flickr. The easiness of acquisition is, however, outweighed by relatively low reliability of position annotations. For Google Street View (created by a single company with quality equipment), Torii *et al.* reported accuracy around 7–15 meters [197]. However, Google Street View restricts the images' locations to streets in cities or the countryside pathways. On the other hand, GPS annotations of images from Flickr and other internet services are unrestricted, but highly unreliable since many different users can manually annotate them. Furthermore, many users shoot their photographs with a diverse spectrum of devices in various uncontrolled conditions.

datset name	class	type	# images	contents	access
Pittsburgh [224]	image	all	62,058	pos + α	download
Pittsburgh250K [197]	image	all	254,064	pos + $\alpha + \beta$	request
Tokyo TM [7]	image	db	98,160	pos + α	request
Tokyo 24/7 [196]	image	db	374,676	pos + $\alpha + \beta$	request
		query	1,125		
IM2GPS small [77]	image	query	237	pos	download
IM2GPS2K [77]	image	query	2,000	pos	download
IM2GPS uniform [77]	image	query	955	pos	download
IM2GPS human [77]	image	query	64	pos	download
IM2GPS3K [207]	image	query	3000	pos	download
YFCC100M [191]	image	all	48,366,323	pos	download
San Francisco [34]	image	db	1,700,000	pos + $\alpha + \beta$	download
		query	803		
VPRiCE 2015	image	all	7,778	match pairs	download
Alps100K [31]	image	all	98,136	pos	download
Quad6K [42]	geometry	all	6,514	pose	download
Dubrovnik6K [114]	geometry	all	6,844	pose	download
Rome16K [114]	geometry	all	16,179	pose	download
Landmark 3D [70]	geometry	all	45,180	pose	download
Cambridge [95]	geometry	all	12,000	pose	download
Landmarks10k [113]	geometry	all	205,162	pose	download
Aachen Day-Night [161]	geometry	db	6,697	pose	download
		query	1,015	N/A	online eval
CMU Seasons [160]	geometry	db	60,937	pose	download
		query	56,613	N/A	online eval
RobotCar Seasons [123]	geometry	db	26,121	pose	download
		query	11,934	N/A	online eval
Symphony Lake [64]	geometry	db	1,409	pose	download
		query	135,966	N/A	online eval
MegaDepth [115]	geometry	all	130,000	pose + depth	download
CH1 [14]	multimodal	all	203	pos + sky	download
CH2 [163]	multimodal	all	948	pos	download

continued on the next page...

. . . continued from the previous page

datset name	class	type	# images	contents	access
Venturi [142]	multimodal	all	3,117	pose + profile	download
CVUSA [216]	multimodal	all	1,500,000	pos + SOP	request
CityScapes [41]	multimodal	all	25,000	pos + seg	download
Kitti [59]	multimodal	all	N/A	pos + var	download
ApolloScape [210]	multimodal	all	N/A	pos + var	download
BDD100K [223]	multimodal	all	†100,000	pos + var	download

† Key frames extracted from 100,000 videos.

Table 3.1: Overview of the visual geo-localization datasets. Abbreviations: query—query images; pos—position; α—yaw, heading angle (around vertical axis); β—pitch, elevation angle (around horizontal axis perpendicular to optical axis); pose—full camera pose (6-DOF); match pairs—matching pairs of images; sky—segmentation of the sky; seg—semantic segmentation annotations; profile—mountain terrain profiles (synthetic terrain silhouettes from DEM); SOP—satellite orthophoto images; var—various annotations are available, usually depth, LIDAR scan, semantic segmentation, street lane segmentation, car instances, etc.; N/A—not available, we were not able to find a valid value.

Pittsburgh datasets. There are several image-based datasets available in the city of Pittsburgh. Google Street View dataset introduced by Zamir and Shah [224] contains 62,058 images acquired automatically from the Google Street View web site, from Pittsburgh, PA, and Orlando, FL. The dataset[1] contains full 360° panoramic images with a distance of about 12 m between consecutive locations. This dataset is suitable for precise localization and camera orientation estimation in urban areas.

Pittsburgh250k, often abbreviated as *Pitts250k* introduced by Torii *et al.* [197], consists of 250,000 perspective images generated from 10,000 Google Street View panoramas from the Pittsburgh area and is available on request.

Tokyo datasets. For Tokyo, the Time Machine dataset has been collected by Arandjelović *et al.* [7] from Google Street View. The authors collected images from various locations in Tokyo throughout the years to leverage the city's changes in appearance. The complete train and validation set contain approximately 100,000 photographs, and the dataset is available on request. Another dataset, Tokyo 24/7, has been created by Torii *et al.* [196]. The authors propose a method to automatically synthesize novel views from Google Street View, given the approximate depth map and panorama segmentation to

[1] https://www.crcv.ucf.edu/projects/GMCP_Geolocalization/

scene planes. The authors synthesized more than two million of views from two hundred thousand street view panoramas. Query images for this dataset were captured manually for 125 locations across Tokyo. Each place has been captured at three different viewing directions and three different times throughout the day, making the query set challenging due to rapid illumination changes.

IM2GPS test sets. In total, three versions of the IM2GPS [77] test set are available. The smallest contains 237 images, the middle one contains 2,000 images, and both are available online[2]. The largest variant of this dataset contains 3,000 images and was also released to public[3] with the revisited version if IM2GPS [207].

YFCC100M: The New Data in Multimedia Research. Thomee *et al.* [191] published one hundred million images in a Yahoo Flickr dataset[4]. The images and videos in this dataset are licensed under Creative Commons, making the data easily usable for anyone. The compressed dataset's metadata consists of 13 GB and contains GPS locations (for 48 million photos and 100,000 videos), tags, timespan, and camera information.

San Francisco Landmark Dataset. Chen *et al.* [34] provided a dataset of 1.7 million street-level images[5] with ground truth labels, geo-tags, and calibration data. Furthermore, a few months after the first part of the dataset, the authors recorded a challenging query set of 803 cellular phone images.

Visual Place Recognition in Changing Environments. VPRiCE dataset[6] for changing environments from VPRiCE challenge 2015 consists of 7,778 images from various outdoor environments and various viewing conditions.

Alps100K dataset. Čadík *et al.* [31] composed a dataset called Alps100K[7]. The dataset contains GPS-tagged images from the area of European Alps collected by querying peak names from Flickr.

[2] http://graphics.cs.cmu.edu/projects/im2gps/

[3] https://github.com/lugiavn/revisiting-im2gps

[4] http://projects.dfki.uni-kl.de/yfcc100m/

[5] https://purl.stanford.edu/vn158kj2087

[6] https://roboticvision.atlassian.net/wiki/pages/viewpage.action?pageId=14188617

[7] http://cphoto.fit.vutbr.cz/elevation/

3.1.2 Geometry-based Datasets

Datasets for large scale SfM and location recognition/pose estimation. Several datasets by Li *et al.* [114] for SfM problems are publicly available online[8]. The largest is the Rome16K and Dubrovnik6K, covering the most famous parts and landmarks of these cities. Also, various smaller datasets for famous landmarks, like Notre Dame Cathedral, Tower of London, Trafalgar Square, Vienna Cathedral, are available.

Quad dataset. Crandall *et al.* [42] also provide SfM datasets. The Quad dataset[9] consists of 6,514 images, about 5,000 images which originate from iPhone 3G contain GPS information, and 348 images contain almost exact GPS coordinates (accuracy about 10 cm.)

Landmark 3D. Hao *et al.* [70] introduced a dataset called *Landmark 3D*[10], which contains 45,000 images of 25 landmarks collected from the Flickr web service. Besides the landmark photos, the dataset also contains reconstructed 3D landmark models. It is suited mainly for landmark recognition.

Cambridge Landmarks dataset. Kendall *et al.* [95] recently published their dataset for 6-DOF camera relocalization using CNN. Train and test images are available online[11], with SfM models used for the camera pose training. The dataset consists of 12,000 images with full 6-DOF camera poses.

Landmarks10k dataset. Another dataset of landmarks and corresponding point clouds reconstructed using SfM has been published by Li *et al.* [113]. It contains more than ten thousand images of famous urban and natural landmarks throughout the world, including the Eiffel Tower in Paris, and the Matterhorn mountain in the European Alps; it is publicly available online[12].

Long-term visual localization. A collection of datasets across varying daytime or seasonal conditions is publicly available[13]. The datasets are collected in urbanized areas and the ground truth is obtained by reconstructing the scenes using SfM approach. Aachen Day-Night dataset [161] captures the differences between day and night at the city of Aachen, Germany. The CMU seasons dataset [160] depicts the urban, suburban, and park scenes in the Pittsburgh, USA, across various seasons. The RobotCar Seasons is a subset

[8] http://www.cs.cornell.edu/projects/bigsfm/

[9] http://vision.soic.indiana.edu/projects/disco/

[10] https://landmark3d.codeplex.com/

[11] http://mi.eng.cam.ac.uk/projects/relocalisation

[12] http://landmark.cs.cornell.edu

[13] https://www.visuallocalization.net/datasets/

of Oxford RobotCar Dataset [123] for autonomous driving collected during one and half year period in a central Oxford, UK using a robotic car equipped with LIDAR, GPS, and Inertial Navigation System (INS) sensors. The Symphony Seasons Dataset was derived by running a SfM reconstruction on a subset of the Symphony Lake Dataset [64], which surveys of a lakeshore over more than three years in Metz, France.

MegaDepth dataset. Li and Snavely [115] created the MegaDepth dataset[14], which contains 100,000 images with reconstructed Euclidean depth data and 30,000 images containing ordinal depth annotations. The authors used modern SfM and multi-view stereo (MVS) [170, 169] methods to reconstruct 3D scenes from the Landmarks10k dataset [113]. In addition to these reconstructions, the authors also propose a series of post-processing steps to prepare the dense depth data for use in deep learning.

3.1.3 Datasets for Methods Using Data of Multiple Modalities

Datasets for horizon-based localization (CH1, CH2). Two datasets for horizon-based localization were published online[15] by Saurer *et al*. [163]. The two datasets contain over 1,000 images with verified GPS position and FOV for every image. For 203 images (CH1 dataset), the horizon segmentation is available.

Venturi Mountain Dataset. Venturi Mountain Dataset [142] is a publicly available[16] dataset of 12 videos from the European Alps region with known ground-truth position and orientation. In total, it consists of 3,117 precisely annotated frames aligned with the terrain model. It is a benchmark suitable for camera orientation estimation algorithms since the dataset contains rotations in all three possible angles (yaw, pitch, roll). However, this dataset is not suitable as a camera localization benchmark because of the small variability of locations (12 unique locations only).

Cross-view dataset. Workman *et al*. [216] introduced CVUSA dataset. It comprises 1.5 million geo-tagged matched pairs of ground level images and an aerial orthophoto map. The authors collected the dataset from Flickr photos and Google Street View, and it can be obtained directly from the authors[17], but it is not available online.

Datasets for city-scale visual geo-localization and scene understanding. Datasets for autonomous driving also gained a lot of popularity in the computer vision research com-

[14] http://www.cs.cornell.edu/projects/megadepth/

[15] http://cvg.ethz.ch/research/mountain-localization/

[16] https://tev.fbk.eu/technologies/venturi-mountain-dataset

[17] http://cs.uky.edu/~scott/research/deeplyfound/

munity. Algorithms for autonomous driving need to solve complex tasks including scene understanding, self-localization and navigation. Datasets developed to tackle these problems usually contain not only camera positions recorded with accurate GPS module, but often include also LIDAR scans of the scene, object detections, semantic segmentation, and stereo images. Acquisition of such datasets is straightforward, since a car equipped with appropriate hardware can easily record hours of the footage while driving in real traffic. One of the first datasets in this category is the Kitti Vision Benchmark Suite [59] recorded in Karlsruhe, Germany. CityScapes dataset added more German cities [41]. ApolloScape [210] recorded in Beijing, China includes also a 3D semantic map of the whole dataset area. One of the largest datasets for autonomous driving, BDD100K [223], has been collected by a crowdsourced platform in the city of New York and the San Francisco bay area, US.

Raw mapping data. Raw mapping multiple modality data are available through USGS[18], where various mapping data like topo maps, aerial photographs, or satellite images are available. The DEM data are available [62] as well. NLCD provides data[19] like land cover attribute maps or tree canopy maps. Maps containing the change between consecutive published versions of land cover maps are available as well.

3.2 Evaluation Methods for Visual Geo-localization

Visual geo-localization methods estimate at least the camera position, and in some instances they also estimate camera orientation or additional parameters, such as focal length [29]. This section introduces the camera pose parametrizations used in this thesis and reviews existing evaluation protocols for visual geo-localization methods that estimate camera position and orientation.

3.2.1 Camera Pose Parametrizations

In this thesis, we use two different camera pose parametrizations. Let us briefly introduce both parametrizations with respective conversions between each other.

[18] http://nationalmap.gov/elevation.html

[19] https://www.mrlc.gov/data

Pinhole camera model in Euclidean space. The first parametrization is a pinhole camera model [72] defined in Euclidean space:

$$P = \mathbf{K}[\mathbf{R}|t] = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & & t_x \\ \mathbf{R}_{3\times3} & & t_y \\ & & t_z \end{bmatrix}, \tag{3.1}$$

where **K** is the intrinsics matrix, **R** is the rotation matrix, and t is the translation of the camera. Camera center C (position) is defined as $C = -\mathbf{R}^{-1}t$, and we use earth-centerd, earth-fixed (ECEF) coordinate system.

Pinhole camera model parametrized by angles The second parametrization is similar to the first one, but instead of rotation matrices and translation in Euclidean space it uses rotation angles:

$$\hat{C} = (\phi, \lambda, h),\ \hat{R} = (\alpha, \beta, \gamma),\ \hat{P} = (\theta, \hat{C}, \hat{R}). \tag{3.2}$$

$\hat{C}$ is the camera position defined by latitude ϕ, longitude λ, and elevation h on an Earth ellipsoid defined by World Geodetic System (WGS84), $\hat{R}$ is the camera orientation defined as a rotation around local axes: yaw α, pitch β, and roll γ. The field of view of the camera is denoted by θ. The camera pose parametrized by $\theta, \hat{C}, \hat{R}$ is denoted by $\hat{P}$.

We can convert one parametrization to the other. To convert from the WGS84 position $\hat{C} = (\phi, \lambda, h)$ to the position $C = (x, y, z)$ in ECEF coordinate system, we use the following closed-form formula:

$$x = [N(\phi) + h]\cos(\phi)\cos(\lambda), \tag{3.3}$$

$$y = [N(\phi) + h]\cos(\phi)\sin(\lambda), \tag{3.4}$$

$$z = [\frac{b^2}{a^2}N(\phi) + h]\sin(\phi), \tag{3.5}$$

$$N = \frac{a}{\sqrt{1 - e^2\sin^2(\phi)}}, \tag{3.6}$$

$$e^2 = 1 - \frac{b^2}{a^2}, \tag{3.7}$$

where a is the semi-major axis, and b is the semi-minor axis defined by the WGS84 ellipsoid. The backward conversion from ECEF to WGS84 is more complicated, and several algorithms exist [24, 233, 222]. The three rotation angles $\hat{R}$ are converted to the rotation matrix **R** as follows:

$$\mathbf{R} = \mathbf{R}_z(-\gamma) \cdot \mathbf{R}_x(-\beta) \cdot \mathbf{R}_y(-\alpha) \cdot \mathbf{R}_{l2w}(\phi, \lambda), \tag{3.8}$$

where $\mathbf{R}_z$, $\mathbf{R}_x$, and $\mathbf{R}_y$ are basic rotation matrices around respective axis:

$$\mathbf{R}_z(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.9}$$

$$\mathbf{R}_y(\phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}, \tag{3.10}$$

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \tag{3.11}$$

and $\mathbf{R}_{l2w}$ is the rotation matrix transforming the local coordinate space to the world coordinate space at the position given by latitude ϕ and longitude λ:

$$\mathbf{R}_{l2w}(\phi, \lambda) = \begin{bmatrix} \vec{e}(\lambda) \\ \vec{u}(\lambda, \phi) \otimes \vec{e}(\lambda) \\ \vec{u}(\lambda, \phi) \end{bmatrix} \tag{3.12}$$

$$\vec{u}(\lambda, \phi) = \begin{bmatrix} \cos(\lambda)\cos(\phi) & \sin(\lambda)\cos(\phi) & \sin(\phi) \end{bmatrix} \tag{3.13}$$

$$\vec{e}(\lambda) = \begin{bmatrix} -\sin(\lambda) & \cos(\lambda) & 0 \end{bmatrix}, \tag{3.14}$$

where $\otimes$ is the outer vector product. Finally, we compute intrinsic parameters $\mathbf{K}$. With the image width I_w, and height I_h, we set $c_x = \frac{I_w}{2}$, $c_y = \frac{I_h}{2}$, and calculate the focal length f from the field of view θ:

$$f = \frac{w}{2\tan(\frac{\theta}{2})}. \tag{3.15}$$

3.2.2 Evaluation Protocols for Visual Geo-localization

Position error. For short distances with positions defined as three dimensional real vectors in the Euclidean space it is enough to measure the position error e_p using the Euclidean distance between the ground truth $C_{gt} \in \mathbb{R}^3$ and the estimated position $C \in \mathbb{R}^3$:

$$e_p(C_{gt}, C) = \|C_{gt} - C\|_2. \tag{3.16}$$

For larger distances across the Earth we parametrize the camera position in WGS84 ellipsoidal geodesic coordinates $\hat{C} = (\phi, \lambda, h)$. The distance between positions on the geodesic ellipsoid needs to be calculated using the Vincenty geodesic distance [204].

Orientation error. We choose to measure the orientation error e_r the same way as the related work [142]:

$$e_r(\mathbf{R}_{gt}, \mathbf{R}) = \arccos\left(\frac{\operatorname{tr}\left[\mathbf{R}_{gt}^{\top}\mathbf{R}\right] - 1}{2}\right), \tag{3.17}$$

where $\mathbf{R}_{gt}$ is the ground-truth rotation matrix, and $\mathbf{R}$ is the estimated rotation matrix. This measures the magnitude of the smallest rotation between the ground truth and the estimated rotation.

The visual geo-localization literature builds on these measures and uses the following methods to plot their systems' overall performance.

3.2.3 Top-*k* Candidates / Percentage of Localized Images

A popular evaluation technique used by state-of-the-art geo-localization methods is the plot of the number of the candidates (horizontal axis) against the fraction of query images from the evaluation set that were localized within the given number of candidates below defined position error [14, 116, 8, 163, 216]. In other words, when the method returns a list of candidate locations, we count how many query images were localized correctly using a fixed number of candidates. We consider the image as correctly localized if at least one candidate out of top-*k* candidates is located within the defined distance ϵ from the ground truth. The curve has ROC-like, non decreasing shape. More formally, with the m candidate positions $C_x(I_j)$ for query image I_j, number N of query images, ground truth position $C_{gt}(I_j)$ of the query image I_j, and the distance threshold ϵ, we calculate the fraction of query images q_m as follows:

$$\forall m \in \{1, ..., k\} : q_m = \frac{1}{N}\sum_{j=0}^{N-1} l(I_j, m), \tag{3.18}$$

$$l(I_j, m) = \begin{cases} 1 & \Leftrightarrow \exists x \in \{1, ..., m\} : e_p\left[C_{gt}(I_j), C_x(I_j)\right] < \epsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{3.19}$$

This method clearly shows how many candidates we must inspect to find at least given fraction of correctly localized query photos. It also illustrates that precise geo-localization is challenging since the methods are often unable to provide adequate localization accuracy for the top-*1* candidate. However, the practical usability of this metric is limited. Usually, the user is interested in the top-*1* candidate, since it is not practical to verify several candidates of possible locations. We can use the following evaluation method to address this problem.

3.2.4 Percentage of Images / Localization (Orientation) Error

Another option is to plot the localization error threshold t (on the horizontal axis) against the fraction q_t of N images from the query set (on the vertical axis) with the same or lower localization error (distance between the estimated position $C(I_j)$ of the image I_j and the ground truth position $C_{gt}(I_j)$) than the threshold t:

$$\forall t \in [0, D_p] : q_t = \frac{1}{N} \sum_{j=0}^{N-1} d_p(I_j, t), \tag{3.20}$$

$$d_p(I_j, t) = \begin{cases} 1 & \Leftrightarrow e_p \left[C_{gt}(I_j), C(I_j) \right] \leq t, \\ 0 & \text{otherwise}, \end{cases} \tag{3.21}$$

where D_p is the maximum localization error. This method was used mainly by global geo-localization methods [77, 213], which retrieve the most probable location (1 candidate) and measure the number of localized queries with the error of the given threshold. Advantage of this evaluation protocol is that we directly observe how accurate is the method for a given fraction of query images from the evaluation set.

Analogically, we can also use this protocol to plot the orientation error. We just change the indicator function d to calculate the orientation error instead of the position error:

$$\forall t \in [0, D_r] : q_t = \frac{1}{N} \sum_{j=0}^{N-1} d_r(I_j, t), \tag{3.22}$$

$$d_r(I_j, t) = \begin{cases} 1 & \Leftrightarrow e_r \left[\mathbf{R}_{gt}(I_j), \mathbf{R}(I_j) \right] \leq t, \\ 0 & \text{otherwise}, \end{cases} \tag{3.23}$$

where D_r is the maximum orientation error in degrees.

3.2.5 Position and Orientation Error per Video Frame

Visual geo-localization methods based on the SfM technique [125, 114, 158, 161], and reviewed Simultaneous Localization and Mapping (SLAM) methods [125, 67, 202, 109] usually evaluate their methods on a per-frame basis. The authors usually present the number of correctly matching query images/frames in the form of a table. Camera position and camera error can be calculated and plotted per video frame. Formally, for each frame I_f of N video frames on the horizontal axis we plot the position (Eq. 3.24) or orientation error (Eq. 3.25) on the vertical axis:

$$\forall f \in \{0,...,N-1\} : e_p\left[C_{gt}(I_f), C(I_f)\right], \tag{3.24}$$

$$\forall f \in \{1,...,N\} : e_r\left[\mathbf{R}_{gt}(I_f), \mathbf{R}(I_f)\right]. \tag{3.25}$$

Alternatively, average position error (Eq. 3.26) and average orientation error (Eq. 3.27) is also often used as a measure of accuracy:

$$\frac{1}{N}\sum_{f=1}^{N} e_p\left[C_{gt}(I_f), C(I_f)\right], \tag{3.26}$$

$$\frac{1}{N}\sum_{f=1}^{N} e_r\left[\mathbf{R}_{gt}(I_f), \mathbf{R}(I_f)\right]. \tag{3.27}$$

Since methods using this evaluation technique are usually verified on the same datasets, it is effortless to compare competitors' method performance. Furthermore, as the methods aim to localize in real-time, the computation time is also a related metric.

3.2.6 Geolocalization Area / Region of Interest

A similar measure to top-k candidates (Section 3.2.3) is the measure of geo-localization area (GA) over the region of interest (ROI, the total area of the search space) [200, 37]. Candidate positions in the search space have assigned their area (which is usually uniform for all the candidates). The candidates are sorted according to the method's confidence. For each query, the GA is calculated as a sum of areas preceding the candidate that contains the ground truth and divided by |ROI|. The graph is plotted for changing GA/|ROI| measure.

Formally, the whole region of interest ROI consists of M particular regions $m_i, i \in \{0, \ldots, M-1\}$ with the area denoted as $|m_i|$. The area of the region of interest |ROI| is equal to the sum of particular areas $|\text{ROI}| = \sum_{i=1}^{M} |m_i|$. Let $m_{gt}(I_j)$ be the ground truth region of image I_j. The method under evaluation assigns a confidence $c(I_j, m_i) \in [0,1]$ to each image I_j and geolocalization region m_i. For the GA/|ROI| threshold t on the vertical axis we plot the fraction of images q_t. We calculate the fraction q_t of N query images by adding $1/N$ for each image that have the sum of the areas lower or equal than the threshold t considering only those regions m_i that have the confidence $c(m_i)$ higher than the confidence of the ground truth region $c(I_j, m_{gt}(I_j))$:

$$\forall t \in [0,1] : q_t = \frac{1}{N}\sum_{j=0}^{N-1} d(I_j, t), \tag{3.28}$$

$$d(I_j, t) = \begin{cases} 1 & \Leftrightarrow \frac{1}{|ROI|}\sum_i |m_i| \leq t, \left\{ i \in \{0, \ldots, M-1\} \;\middle|\; c(I_j, m_i) > c[I_j, m_{gt}(I_j)] \right\}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.29}$$

In the case all the candidates have an equal area, this method is the same as top-k candidates. The method would be more informative for a non-uniform sampling of the search space since it would penalize wrong estimations with a large area.

3.2.7 Precision / Recall

Precision/recall is a measure of relevance used for the evaluation of classification, place recognition, and retrieval methods [8, 127]. We avoid using precision/recall in this thesis, since distance based measures are more suitable for evaluation of geo-localization methods, but we include it in this survey for completeness. Researchers usually plot the recall on the horizontal axis and the precision on the vertical axis. Formally, a retrieval system retrieves $M \in \mathbb{N}$ documents from a database containing $N \in \mathbb{N}$ relevant documents. We define a precision P and recall R as follows:

$$P = \frac{N \cap M}{M}, \quad R = \frac{N \cap M}{N}. \tag{3.30}$$

3.3 Chapter Summary

We presented an overview of datasets and evaluation practices in visual geo-localization. We divided the datasets based on the acquisition technique. *Image-based* datasets are composed of visual material containing location metadata, *e.g.*, GPS position stored in EXIF, which could be acquired using hardware solutions or annotated manually. *Geometry-based* datasets add further processing of such raw data and use photogrammetry methods to estimate relative camera motions to deliver 3D scene reconstruction and full camera pose information. Alternatively, geometry-based datasets may be also collected using specialty hardware, *e.g.*, laser scanners, such as LIDAR. While *image-based datasets* merely contain a rough estimate of the camera rotation, *geometry-based datasets* usually contain fairly accurate camera rotation information.

We reviewed several datasets targeted on a cross-domain scenario, which we called *datasets of multiple modalities*. *Datasets of multiple modalities* do not have any unified acquisition technique. Urban datasets focused at solving problems for autonomous driving are usually collected relatively easily in real traffic using a vehicle equipped with GPS, INS, LIDAR, and 360°cameras, or calibrated camera rigs. On the other hand, datasets for outdoor natural environments, such as mountains, are much more difficult to collect. Accessing mountainous areas is difficult—majority of mountainous areas cannot be simply accessed by a car, but need to be visited by passionate tourists and hikers. The challenge also resides in the significant appearance gap between different modalities, *e.g.*, between the photograph and rendered DEM model; correspondences between such modalities are often tricky to annotate manually, even for human experts. These difficulties may explain the small number of datasets for cross-domain visual geo-localization in natural environments. From all the reviewed datasets, only a small handful contains images of natural environments. Specifically, images relevant to our work may be found only in CH1 [14], CH2 [163], Venturi Mountain dataset [142], and in a part of the MegaDepth [115] dataset.

In the second part of this chapter, we introduced two camera pose parametrizations we use in this thesis and reviewed standard evaluation methods in the context of visual geo-localization. Usually, errors in position and rotation estimates are measured and aggregated based on the number of allowed candidate solutions, or distance from the ground truth position.

Part II

Datasets for Visual Geo-localization in Outdoor Environments

Developing novel computer vision methods depends on datasets, which we use for method evaluation and comparison. Moreover, in the last decade, the training data is also indispensable for optimizing methods' parameters using deep learning. However, as we showed in the state-of-the-art overview, a limited selection of datasets is appropriate for solving single image localization based on comparing an input photograph and a synthetically rendered terrain. In this part, we present methods and datasets we developed to overcome this issue.

Chapter 4

GeoPose3K: Mountain Landscape Dataset for Camera Pose Estimation in Outdoor Environments

Visual geo-localization in outdoor environments is often based on the comparison of a query photograph with a DEM model. Comparing such different modalities is challenging and has been the subject of several studies in the last two decades [190, 189, 186, 146, 133, 14, 69, 200, 37, 163]. These studies often build their algorithms with handcrafted features, *e.g.*, horizon lines, silhouette edges, keypoints, and semantic segmentation. In the last decade, researchers focused on developing more complex data-driven predictive models for feature extraction, which significantly improved almost every area of computer vision. To allow predictive model training for our specific domains—photographs and rendered DEMs, we introduce a novel dataset of photographs precisely aligned with the terrain. Our dataset contains full camera poses (positions and orientations) and various metadata rendered directly from the terrain model.

Contributions. We present a new dataset *GeoPose3K* which addresses three main issues with existing datasets for camera pose estimation in natural, mountainous environments: (I) a small number of images with verified ground truth position, (II) an absence of full camera orientation and (III) an absence of metadata for the training and evaluation of feature detectors and further applications outdoors. The proposed dataset *GeoPose3K* consists of more than three thousand photographs collected mainly from the photo sharing site Flickr.com. All photographs originate in the Alps region, which is the highest mountain range in Europe. For each image all camera pose parameters (GPS position, FOV, full orientation) are provided. The camera pose parameters were assessed with an image-to-model matching technique [15] and manually verified. In order to enable the training and development of future approaches for outdoor environments, we provide various synthetic data per image: depth map, normal map, simulation of illumination during the day, and semantic labels. One sample image from our dataset with corresponding synthetic data is shown in Fig. 4.1.

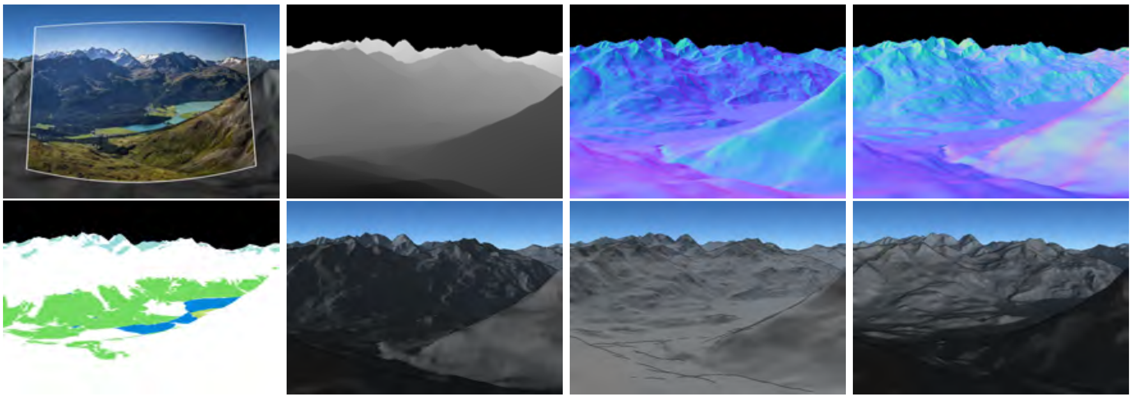


Figure 4.1: *GeoPose3K* dataset: for each mountain landscape photograph, the dataset contains (in reading order) its GPS coordinate and camera orientation, distance from the camera in meters, normals w.r.t. camera, normals w.r.t. cardinal direction, semantic labels and approximate illumination during the day (here shown at 5am, 12pm and 8pm).

4.1 Related Work

Geo-localization datasets for natural environments Porzi *et al.* published the Venturi Mountain Dataset [143] with annotated camera poses for video frames. It contains 3,117 video frames from 12 video sequences. For this reason the Venturi dataset contains a lot of similar images: while it is suitable as a benchmark for camera orientation estimation, it is not a suitable benchmark for geo-localization problems. An image-based dataset from the Alps region called Alps100K was collected by Čadík *et al.* [31]. It was downloaded from Flickr by querying hill names and filtering out evident outliers using CNN. The photos in this dataset contain a GPS position, elevation and FOV. However, the dataset does not provide *camera pose* parameters and the provided ground truth *geo-locations* were not verified and thus might be noisy. Datasets for visual geo-localization called CH1 and CH2 were provided by Saurer *et al.* [163]. Both datasets contain in total a thousand images with known ground truth GPS location and FOV, but *camera orientation* is not provided. Segmentation of the sky and foreground terrain is provided for 203 images in the CH1 dataset.

RGB-D datasets Since we provide additional synthetic metadata as depth and semantic labels, etc. (see Fig. 4.1), we also briefly overview works introducing existing datasets containing similar data. Thanks to the ease of RGB-D images acquisition using devices such as Microsoft Kinect, many indoor RGB-D datasets exist [54]. Acquisition of outdoor RGB-D datasets is more challenging, because the depth range and resolution of depth sensors is limited. Saxena *et al.* [164, 165] used a laser depth scanner with a maximum depth of 81 m and resolution 55 × 305; Kitti dataset [59] contains 3D point clouds collected by a LIDAR sensor. However, such approaches are unusable for mountainous environments, where the depth of the scene varies from several meters to hundreds of kilometers. An option suitable for mountainous environments would be to calculate depth from two stereo images, but the disparity needed to obtain viable results would be prohibitive for practical scenarios. We solve these problems by rendering the corresponding depth for each image from a DEM.

Semantic segmentation datasets Several standard datasets for semantic segmentation exist [50, 179, 129, 118]. The methods and datasets for semantic segmentation are usually generic—they contain a number of classes that are supposed to cover various kinds of content. While existing datasets, such as Pascal-Context dataset [129] contain relevant classes for mountainous areas—*mountain*, *rock*, *tree*, *grass*, *water*, *road*, *snow* or *sky*, it does not provide mountain specific classes—*forest*, *glacier*, *cliff* or *moor*. With this motivation, we include synthetic semantic labels into *GeoPose3K* dataset. We overlay the DEM with the 13 most relevant OpenStreetMap natural features[1] and for each image in the dataset render a corresponding synthetic view containing the semantic labels (see Fig. 4.1).

An approach to camera orientation estimation in outdoor scenes based on semantic segments was published by Baatz *et al*. [12]. However, they used only four classes – residential area, bodies of water, sky, and "everything else". Also, their dataset seems to only contain several images and is not publicly available.

4.2 Dataset Acquisition

We introduce *GeoPose3K*, a dataset of images with known parameters—camera field of view θ, camera position $\hat{C} = (\phi, \lambda, h)$ (latitude, longitude, elevation), and camera orientation $\hat{R} = (\alpha, \beta, \gamma)$ (yaw, pitch, roll, see Eq. 3.2). The dataset consists of two main parts. The first part contains 339 images captured and annotated manually (over 10% of the whole dataset). For each image the GPS position was recorded by its authors using a GPS sensor. We found camera orientation ground truth by selecting correct correspondences of the image and the DEM similarly to Kopf *et al*. [99]. However, such manual collection and annotation is a lengthy and tedious task. Therefore, we processed the second part of the dataset (2,772 images) using a semi-automatic algorithm outlined in Sec. 4.2.1. The whole dataset consists mainly of images from an online photo service. We also assessed orientations for the CH1 dataset images which initially contained only the camera position and field-of-view.

[1] http://wiki.openstreetmap.org/wiki/Key:natural

We used photographs with a known FOV θ and GPS position $\hat{C} = (\phi, \lambda, h)$ from the Alps100K dataset by Čadík *et al.* [31], who originally acquired the images from the Flickr online sharing service. Therefore, we assume the parameters $\theta, \hat{C}$ to be known but noisy. Our goals were: (I) for each image I, recover a correct position and estimate camera orientation $\hat{R} = (\alpha, \beta, \gamma)$ so that we can assemble the complete camera pose $\hat{P}(I) = (\theta, \hat{C}, \hat{R})$; (II) classify each recovered camera pose as *viable* or *incorrect*; and (III) refine parameters of each *viable* camera pose $\hat{P}$. We assume the camera pose of a given image to be *viable* if a human user observes apparent correspondence between the query image and the synthetic image rendered from DEM with given camera pose parameters (see Fig. 4.2). We should note that this does not necessarily mean the camera pose is 100% correct; for this reason, we pass *viable* images to the refinement process. An *incorrect* camera pose means there is no apparent correspondence to the DEM; we discard images with an *incorrect* camera pose. We included images that passed the refinement process into the dataset with the best camera pose we found.

4.2.1 Method Outline

Since we know the camera position (ϕ, λ) for each image, we recover the elevation h by querying the DEM at the position. For camera orientation estimation, we used an approach based on the *Alignment Metric* proposed by Baboud *et al.* [15]. We propose an improvement of their *Alignment Metric* for camera orientation estimation and show that it performs better than the baseline. We have used the improved *Weighted Alignment Metric* to automatically estimate the camera pose of 30,000 photos from the Alps100K dataset. We manually verified the estimated camera pose of each photo. In case the found camera pose was *viable*, we added the photo into a list of candidates.

For each photo in the list, which consisted of more than 3,000 candidates, we sampled several hypotheses of the FOV and their position around the original FOV θ and position (ϕ, λ). We sample the position to mitigate the positional error introduced by an imprecise GPS tag; we sample the FOV to eliminate possible inaccuracies of the recorded focal length or camera sensor size. The FOV might be incorrect due to several factors. First, the image might be cropped (many images in the Alps100K dataset contained artistic borders, which were cropped automatically). Secondly, the FOV might be wrongly calculated due to an incomplete list of cameras and their sensor sizes. Finally, users and third-party software might manipulate values stored in EXIF before online sharing. One should note that moving the camera while keeping the original FOV fixed is not equivalent to adjusting the FOV with a fixed position. According to Hartley and Zisserman [72], 3D scenes containing objects near the camera are perspectively distorted. Hence moving the cam-

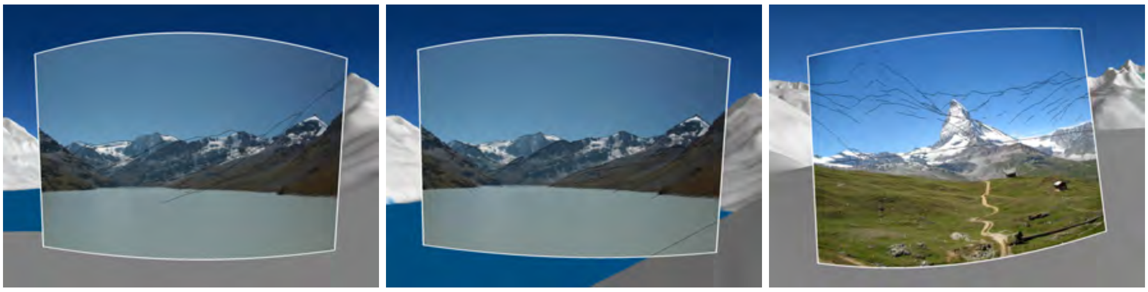


Figure 4.2: Example of a *viable*, but contaminated camera pose (left), refined camera pose (middle) and *incorrect* camera pose (right). Synthetic mountain silhouettes are overlaid with the aligned image. Image credit: left and middle image—Flickr.com user *Michael Holtrop*: https://www.flickr.com/photos/bartje_assen/2851555201/, right image—Flickr.com user *Bossi*: https://www.flickr.com/photos/thisisbossi/2973222425/

era towards/backward a nearby mountain will change the perspective distortion, while a decrease/increase of FOV (zoom in/out) does not affect it.

We reran the camera orientation estimation method for each sampled hypothesis and manually chose the camera pose that visually matched the DEM best. Only if the resulting camera pose matched the DEM precisely (we tolerate error up to several pixels), we added it into the dataset.

4.2.2 Alignment Metric

For each possible camera orientation (α, β, γ), the original method by Baboud *et al.* [15] calculates the image-to-DEM matching score using edges from a query image and silhouettes extracted from the panoramic rendering of a digital terrain model. This score was designed for calculating a precise camera orientation given a known position. Using this score to verify more (possibly incorrect) position candidates is difficult, since the absolute value of the score varies between positions due to differences in detected edges (different lengths and shapes). The found camera pose has to be visually inspected by a human user to recognize a *viable* or *incorrect* result. Let us briefly review the *Original Alignment Metric* and then propose our improved version, *Weighted Alignment Metric*.

Original Alignment Metric

We reformulate the problem of matching as introduced by Baboud *et al.* [15], who proposed a matching score per edge es (4.1):

$$es(e, D) = \sum_{j \in e}^{|e|} |c(e_j, D)| \left(\frac{1 + c(e_j, D)}{2} d(e_j)^p + \frac{1 - c(e_j, D)}{2} m \right), \tag{4.1}$$

and the final matching score $s(Q, D) = \sum_{e \in Q} es(e, D)$, where Q denotes the set of edges extracted from a query image, and D denotes the set of synthetic silhouettes with which the query image is matched. The term $d(e_j)$ measures the length of the edge segment e_j, p and m are constant parameters. The parameter p defines non-linear weighting of edges based on their length, and the negative parameter m defines the cost of edge crossings.

The term $c(e_j, D)$ measures the spatial configuration of a query edge segment e_j concerning a synthetic silhouette segment e_i. In case the edge segment e_j is parallel with silhouette segment e_i, the term is equal to 1, and in case the edge segments are crossing each other, the term $c(e_j, D)$ is equal to -1, and to 0 in other cases. Two edge segments are parallel if all points of the query edge segment e_j are in the $\mathcal{E}$ neighborhood of the synthetic silhouette segment e_i. In summary, the score $s(Q, D)$ sums up the lengths of edges parallel with some synthetic silhouettes and penalizes edges crossing the synthetic silhouettes.

Weighted Alignment Metric

In the original alignment metric (4.1), all edges were assigned the same importance regardless of their visual appearance, even though their appearance can correlate with their importance for matching. To improve the original method's matching performance, we propose to weight image edges based on their strength. We implemented edge strength as a weight of the edge segment $w(e_j) \in \langle 0, 1 \rangle$. Weight w is multiplied with terms $d(e_j)$ and m in (4.1) respectively, so we get:

$$es(e, D) = \sum_{j \in e}^{|e|} |c(e_j, D)| \left(\frac{1 + c(e_j, D)}{2} w(e_j) d(e_j)^p + \frac{1 - c(e_j, D)}{2} w(e_j) m \right). \tag{4.2}$$

Weighted Edge Detector

To detect edges from query images with meaningful weights, we adopt the edge detection framework by Dollár and Zitnick [45]. Their approach predicts a 16×16 edge map from a larger 32×32 image patch. Individual predictions are averaged to produce a soft edge map for the whole input image. The learning problem is solved using structured random forests. In order to use standard node splitting criteria, the structured space of labels $\mathcal{Y}$ is mapped to a discrete set of labels C by a two-stage mapping via an intermediate space $\mathcal{Z}$ at each node. The authors assume segmentation maps being available for training. Instead, we use our synthetic depth maps. To use depth maps as labels, we redefine the intermediate mapping $\Pi : \mathcal{Y} \rightarrow \mathcal{Z}$ to produce a vector that encodes depth difference $y(j_1) - y(j_2)$ for every unique pair of indices $j_1 \neq j_2$ within a label patch $y \in \mathcal{Y}$. In practice, we sample $m = 256$ dimensions of $\mathcal{Z}$, resulting in a node-specific reduced mapping Π_Φ, further discretized as in the original paper [45].

4.2.3 Candidate Refinement

We added images for which the *Weighted Alignment Metric* recovered a *viable* camera pose into a list of candidates. The camera pose could be contaminated due to a combination of many factors: an imprecise GPS tag, imprecise FOV, imprecise DEM, and the distortion of a query image. Since we obtained a high number of images with *viable*, but contaminated camera pose after the camera pose estimation process, we further refined contaminated camera poses.

We hypothesized a position with eight samples regularly placed around the original position (ϕ, λ). We placed four samples in the corners of a smaller square with a side of 500 m, and four samples in the corners of a larger square with a side of 1,000 m. The original position (ϕ, λ) is the center of both nested squares. For each new position, we have also sampled FOV θ of the camera. The minimum value of FOV was $\theta - 0.1\theta$, the maximum $\theta + 0.1\theta$, and there were in total four steps sampled linearly between the minimal and maximal value. We ran our *Weighted Alignment Metric* on each sampled position. In this way, we obtained thirty-two new camera poses for each candidate. Finally, an expert user manually verified these camera poses. In case the best camera pose of the candidate was precise enough (the alignment error was not greater than several pixels, see the middle image in Fig. 4.2), the new refined camera pose was added into the dataset.

The process of candidate refinement was very demanding on computational resources and time. The alignment of all sampled positions and FOV's took three weeks on seven computers equipped with Intel Core i3-4360 CPU and NVidia GTX 980 GPU. In addition, it took one person-month to assess the estimated camera poses manually.

4.3 Synthetic Data

Each image I in the dataset is provided with a camera pose $\hat{P}(I) = (\theta, \phi, \lambda, h, \alpha, \beta, \gamma)$. This camera pose allowed us to support the dataset with additional synthetic data rendered from DEM.

Depth. We acquired a depth map by pixel-wise raycasting and measuring the distance from the camera to the first intersection with scene geometry. The depth map accuracy depends on the DEM resolution; our DEM consists of samples spaced by 24 meters, and we obtained it from the viewfinderpanoramas website[2].

Normals. We produced two types of normals. We calculated normals w.r.t. the camera relative to the camera position and orientation; we calculated normals w.r.t. cardinal direc-

[2] http://viewfinderpanoramas.org

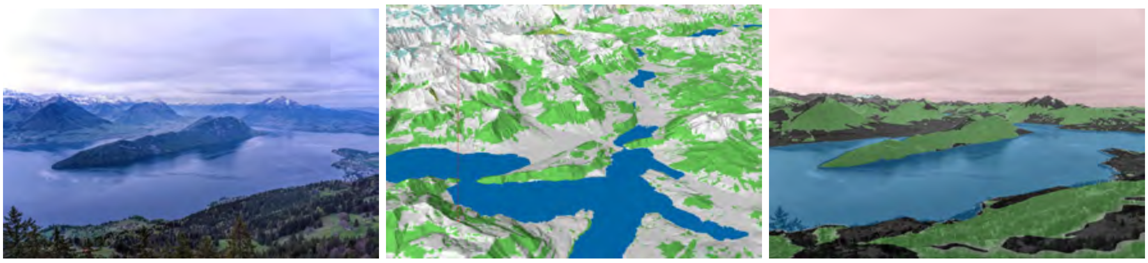


Figure 4.3: Example of OpenStreetMap semantic segments provided per dataset image. Left: original photograph. Middle: terrain metadata from OpenStreetMap [68] rendered on the digital elevation model. Right: original image overlaid with terrain metadata from OpenStreetMap. Color coding: **sky**, **water**, **forest**, glacier, rock, **other**.

tion relative to the world coordinate system. Original normals of the surface $n \in \mathbb{R}^3$, where $n_x, n_y, n_z \in \langle -1, 1 \rangle$, are encoded into the RGB image $n'_{rgb} = 0.5 + 0.5 \cdot n$, where $n'_r, n'_g, n'_b \in \langle 0, 1 \rangle$.

Illumination. Illumination approximation was simulated hour by hour from 4 am till 9 pm on 21$^{\text{st}}$ June, when the days are the longest during a year. We calculated the illumination simulation using a local illumination model, so that it does not contain casted shadows.

Semantic Segments We use publicly available metadata from OpenStreetMap [68]; however, other sources (*e.g.*, NASA Visible Earth[3] or USGS Land Cover[4]) are usable, too. More specifically, we render 13 natural and physical land features from OpenStreetMap natural feature set[5]: bare rock, cliff, fell, forest = wood, glacier, grassland, moor, scree, shingle, sinkhole, and water. We map each feature on one color layer in a geo-referenced texture. Subsequently, we drape the texture on our 3D terrain model (see Fig. 4.3, middle). Assuming the image (*e.g.*, Fig. 4.3, left) is correctly aligned with the model, we can project the texture onto the image using the virtual camera while correctly accounting for the visibility thanks to the 3D terrain model. This procedure results in the final pixel-wise semantic labels (Fig. 4.3, right).

4.4 Dataset Properties

The *GeoPose3K* dataset consists of two main parts, collected *manually* and *semi-automatically*. The first part contains 339 images, which were collected and annotated manually. Since this was a tedious task, collecting more dataset samples in this way was unfeasible. The

[3] http://visibleearth.nasa.gov/view.php?id=61004

[4] http://www.usgs.gov

[5] http://wiki.openstreetmap.org/wiki/Key:natural

second part of the dataset contains 2,772 images for which we optimized the camera parameters using *Weighted Alignment Metric* combined with hypotheses sampling and manual selection of the best candidate, as was described in Sec. 4.2.3.

4.4.1 Potential Bias for Evaluation of Camera Orientation Estimation Methods

We gathered the second part of the dataset *semi-automatically*, with the help of a method by Baboud *et al*. [15]. For this reason, methods based on edge features might be potentially privileged over algorithms based on different principles. This bias must be taken into account when using GeoPose3K for the evaluation of orientation estimation methods. However, for evaluating problems based on different features, the usage of the dataset is valid. We illustrate this property in a benchmark evaluating a state-of-the-art horizon-based geo-localization method by Saurer *et al*. [163] in Sec. 4.5. Our evaluation shows that the dataset difficulty on the localization task is on-par with datasets collected solely manually (CH1, CH2).

For evaluating methods similar to the method by Baboud *et al*. [15], a *manually* collected part of the dataset (339 images) shall be used. Since selecting the images in this part was not affected by any algorithm, there is no limitation on which methods can be evaluated using this part of the dataset.

4.4.2 Statistics

The majority of images in the GeoPose3K dataset originate from the Alps100K dataset. The GeoPose3K dataset consists of images containing an accurate GPS tag and a reasonable portion of the mountain scene allowing the registration with the terrain model. This definition restricts the set of images in which we can generalize our conclusions. However, the GeoPose3K dataset has adequate coverage by users—two thousand users have a single photograph in the dataset, around a hundred and fifty users have two photographs, and only a single user has twelve photographs in the dataset, which is the largest number of photographs created by a single user. The majority of images in the dataset were taken between 2007 and 2014. The user and year distributions exhibit some degree of similarity to the Yahoo Flickr Creative Commons 100 Million dataset (YFCC100M) [191]. We employed a two-sample Kolmogorov-Smirnov hypothesis test with a null hypothesis: the GeoPose3K and YFCC100M distributions do not differ significantly. For the user distribution, the two-sample Kolmogorov-Smirnov test failed to reject the null hypothesis ($D(6487, 12) = 0.3042, p = 0.18$). For yearly distribution, the same test clearly rejected the null hypothesis ($D(35, 35) = 1, p < 0.001$). From this we can conclude that year's distributions differ significantly, but we could not prove the same for the user distribution. This fact illustrates that the GeoPose3K and YFCC100M share some degree of similarity, but GeoPose3K is a more specific subset than YFCC100M.

Despite of the above-defined restrictions, the GeoPose3K dataset enables us to speculate about the following questions. What cameras are the most common in the mountain environment, and which of them are the best candidates for visual geo-localization? Which focal lengths are the most successful for matching mountain images? Are people taking photographs with a zero roll angle? Which orientation at what time is the most favorite? To answer these questions and to illustrate the properties of the *GeoPose3K* dataset, we measured several statistics.

Geographical Distribution. We visualized the distribution of *geo-locations* of photographs in the *GeoPose3K* dataset in Fig. 4.4 (left). We built the dataset from Alps100K images, and hence, the photos are distributed over the whole Alps region. However, the photos are not distributed uniformly—the areas of tourist interest like Switzerland and northern Austria contain more images than other areas.

Cameras. Due to its excessive size, we attached a full list of cameras in the Appendix, in Table A.1. The most frequent camera in the dataset is Canon DIGITAL IXUS 860 IS (around 8% of dataset images). Interestingly, the first and third most frequent cameras in our dataset are not equipped with a built-in GPS sensor; according to this fact, at least 10% of the dataset images obtained their original GPS coordinates by a third-party logger or were geo-tagged manually.

Focal Lengths. A histogram of focal lengths recalculated to 35mm equivalent is visualized in Fig. 4.4 (middle). Shorter focal lengths are typical for mountain landscape images (Alps100K), and the measured distribution of *GeoPose3K* shows this bias as well.

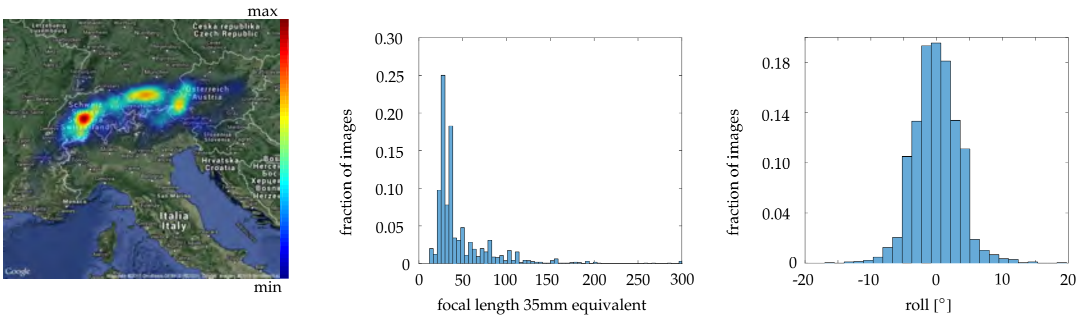


Figure 4.4: Dataset statistics. Left: distribution of photo locations in *GeoPose3K* dataset. Middle: histogram of focal lengths in the dataset. Right: average camera roll.

Average roll. People usually aim to level their photos with the horizon line. Accordingly, the histogram of roll angles in Fig. 4.4 is centered around zero. However, keeping the camera level may be difficult in the mountains, and some shots are rotated. This fact justifies that the geo-localization methods need to be optimized for a camera roll as well.

Time-Orientation Correlation. According to Fig. 4.5 (right), photos in the *GeoPose3K* dataset were taken most frequently around 2 pm with a heading of 90°. In general, photos were photographed mainly between 10 am, and 4 pm and the favorite headings ranges are between 0°-120° and 170°-300°.

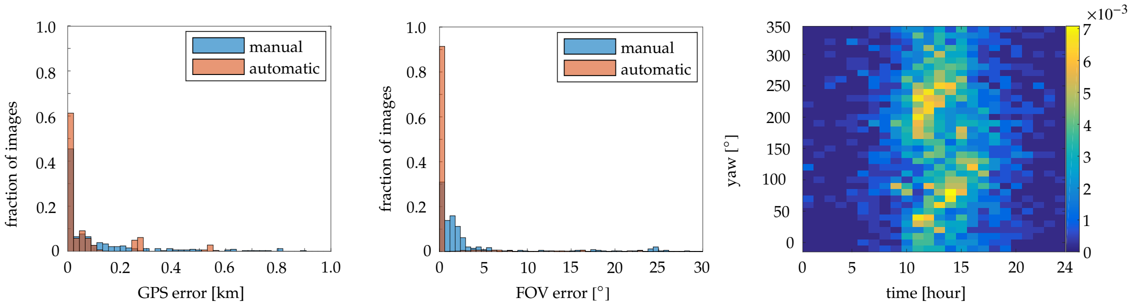


Figure 4.5: Dataset statistics. Left: histogram of GPS error distribution. Center: histogram of FOV error distribution. Positions and FOV's refined using our semi-automatic method are drawn in brown, and the positions and FOV's refined manually are in blue. Right: time/orientation correlation visualization.

GPS and FOV error. We measured GPS and FOV errors of manually and automatically (Sec. 4.2.3) refined dataset images. We measured geo-distance between the original and the refined GPS position for each image using the Vincenty algorithm [204] and plotted a histogram of these errors (see Fig. 4.5 on the left). We measured a similar histogram of FOV errors (Fig. 4.5 in the middle), based on the angular distance between the original and the refined FOV. According to our measurements, there are discovered discrepancies in GPS values. The images sometimes exhibit noisy GPS tags, probably due to manually edited geo-tags, lousy reception of a GPS sensor, or the fact that cameras have their GPS refresh rate set to a long time interval. The FOV error histogram peak is near zero, reflecting that the original fileds-of-view of photos in the dataset were nearly correct. Imperfections in FOV up to 1°–2° discovered by the manual annotation are probably caused by tiny inaccuracies of the digital elevation model or by a tiny GPS error. Therefore, the FOV error within a small margin of 1°–2° is assumed to be correct.

Edge accumulation. The *GeoPose3K* dataset allows us to analyze properties of the query edges and discover their importance. For the following experiment, we sampled a grid around each query image, with the query image located at the center of the grid. The grid had 9×9 samples and the distance between the consecutive samples was 0.001° in both North-South and West-East directions. We rendered the synthetic silhouettes from each grid location and matched them to the query image edges. We incremented 1 to all pixels in the accumulator containing a synthetic silhouette, which contributed positively to the matching score. Finally, we ran such an evaluation for every position in the grid and summed up the accumulators to obtain one accumulated image for a query. We visualized an example of such an accumulated image in Fig. 4.7.

To analyze the importance of edges in all images in the dataset, we have created the average accumulated edge map (Fig. 4.6, left). The most populated one is the central area of an image with a slight bias towards the frame's bottom. Similarly, we have created an average accumulated cylindric panoramic image, where we accumulated each image according to its original camera orientation (Fig. 4.6, right). As one would expect, the area around the horizon is the most populated; however, the silhouettes off ±10° seem to be of a similar importance. The image further exhibits a good *GeoPose3K* dataset coverage of camera orientations.

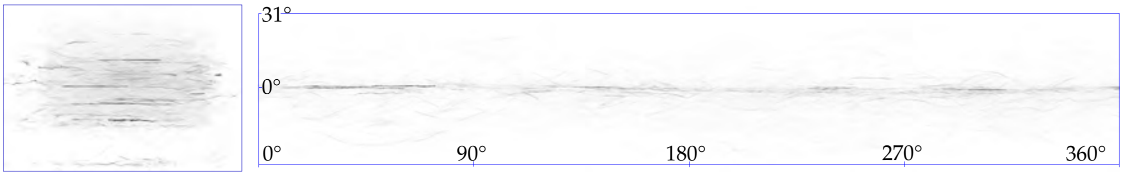


Figure 4.6: Edge accumulation. Left: Normalized average from all accumulated images. Right: Accumulated panorama image.

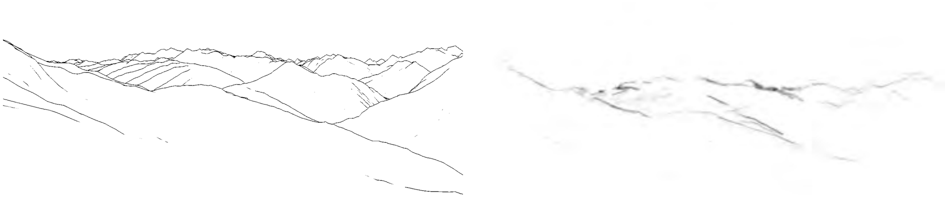


Figure 4.7: Accumulation of matched edges for a single query image. We split the area around the GPS location of the query image uniformly into a grid of 9×9 cells (resolution of 0.001° in both N-S and W-E directions). From each cell we render synthetic silhouettes (**left**) and match them to edges in the query image. In the accumulated image, we increase the value of all pixels containing a synthetic silhouette, which contributed positively to the matching score. We run such an evaluation for every cell in the grid and sum up the accumulators to obtain one accumulated image for a query (**right**).

4.5 Experiments

We evaluate the performance of the *Original* and the *Weighted Alignment metric* in Sec. 4.5.2. We show that the *Weighted Alignment metric* outperforms the *Original Alignment metric* by a large margin, which allowed us to build the dataset more efficiently.

Furthermore, we use the *GeoPose3K* dataset to extensively evaluate the state-of-the-art method by Saurer *et al*. [163] for horizon-based visual geo-localization in the mountains in Sec. 4.5.4. By this evaluation, we bring more profound insight into the dataset properties; according to our measurements, the difficulty of the new *GeoPose3K* dataset for visual geo-localization is higher than the difficulty of the original CH1 [14] dataset, and is similar to the difficulty of the CH2 [163] dataset. *GeoPose3K* allows us to evaluate the baseline using more photos covering a larger area than the original datasets. For this evaluation, we issued the vastest area ever—our most massive experiment deals with an area more than twice the size of the original area reported by Saurer *et al*. [163]. Additionally, *GeoPose3K* also allows us to evaluate the camera heading accuracy. The evaluated method [163] is capable of camera heading estimation; however, the original paper [163] did not evaluate it quantitatively, because there was no suitable dataset containing *camera orientations*. Furthermore, we evaluated geo-localization performance using three fully automatic horizon line extraction methods to illustrate that the automatic horizon line extraction is still challenging.

4.5.1 Setup

The performance of the *Original* and the *Weighted Alignment Metric* was evaluated by manually counting correctly registered images. As a test set, we randomly selected 400 images from the Alps100K test set [31]. The number of selected images is based on the fact that the alignment metric is demanding on computational time and validation demands many human resources, and testing several variants of edge maps would be too expensive with a broader test set.

For experiments with camera localization and orientation estimation, we reimplemented the state-of-the-art method by Saurer *et al*. [163] and measured its performance on *GeoPose3K*. The method utilizes a database of densely sampled horizon lines from a DEM to retrieve locations given a query horizon line. We extracted a database of synthetic horizon lines that covers 86,000 km^2 (red area in Fig. 4.8(a)—*GP1*), which is more than twice the size of the area used in the original paper [163] (40,000 km^2). We sampled the area of interest in both N-S and W-E directions with a resolution of 0.001°. Samples in N-S and W-E directions are 111 m and 78 m far from each other, respectively.

Scenario	**Thresholded**	**Weighted**
Compass	2.75%	2.75%
CannyDM	6.20%	0.01%
Silhouette	**7.25%**	**9.75%**

Table 4.1: Image registration experiments: The table shows a fraction of successfully registered images from 400 randomly selected images from Alps100K [31] test set.

The original method facilitates the bag-of-words retrieval adapted to horizon line contours. The approach allows us to retrieve the approximate heading and position of the camera using a voting scheme. We used voting for location and direction with 2.5° and 10° descriptors and 3° directional bin size, which seemed to be the best choice according to the original paper's results. We used the evaluation method proposed by Baatz *et al.* [14]; we measured the distance between a candidate and a ground truth location, assuming the location is correct if the distance is smaller than 1 km. Finally, we plotted the cumulative percentage of correctly localized images given top-*k* candidates.

4.5.2 Performance of the Alignment Metric

Both the *Original* and the new *Weighted Alignment Metric* (Sec. 4.2.2) assume edge maps on their input. We experimented with several edge map acquisition methods, including novel depth-based approaches (described below), to find the best possible settings. We summarize the results in Table 4.1 and show that the weighted variant of the silhouette detector is by far the best.

Thresholded edge maps. We used a *Compass* edge detector [156] in the baseline metric [15]. Similarly to the authors, we thresholded the edge map ($\tau = 0.7$) to keep only significant edges. In Tab. 4.1 we denote this edge detector as *Compass* | *Thresholded*. We also experimented with an alternative approach: the thresholded *Canny detector applied on the depth map estimate* (see *CannyDM* | *Thresholded* in Tab. 4.1). We estimated the depth map using a dark channel prior directly from an input image [79]. The edges were then obtained from this depth map using a thresholded Canny edge detector ($\tau = 0.25$), representing depth discontinuities. The resulting edge map often exhibits more distinctive edges, especially the more distant ones, compared to the edges detected directly from the original query image. Additionally we used a thresholded variant of the *Weighted edge detector* described in Sec. 4.2.2. We kept only edges with weight exceeding the threshold $\tau = 0.1$, to detect the strong edges and neglect the background noise. In the Tab. 4.1 we denote this variant as *Sihouette* | *Thresholded*. All thresholded variants have all edges with a unit weight.

Weighted edge maps. We have included weighted versions of both methods (*Compass* and *CannyDM*) described above, where the raw edge strength was linearly rescaled into edge weights $w \in \langle 0, 1 \rangle$ (see *Compass* | *Weighted* and *CannyDM* | *Weighted* in Tab. 4.1). Finally, we experimented with the *Weighted edge detector* described in Sec. 4.2.2, see *Silhouette* | *Weighted* in Tab. 4.1. Our measurements show that the weighted variant of our matching metric produces significantly better results, and the new weighted silhouette detector is the preferred edge map construction method.

4.5.3 Automatic Horizon Line Detection

The baseline localization method requires a horizon line as a query input. To measure the method's performance on the *GeoPose3K* dataset, we experimented with several algorithms for the automatic detection of horizon lines.

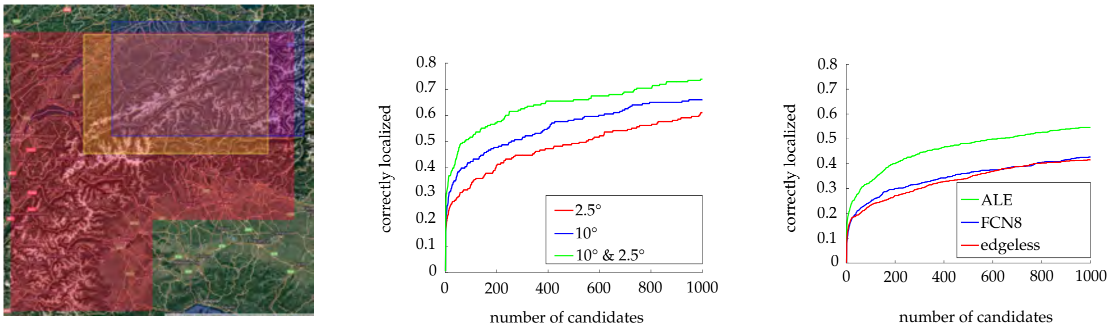
Automatic Labeling Environment (ALE) [106]. ALE is an energy minimization-based semantic segmentation framework adopted for sky extraction by Saurer et al. [163]. Specifically, the energy is predicted by a pixel-wise classifier trained on contextual and superpixel feature representations. The method uses multiple bag-of-words representations over the random set of 200 rectangles and superpixels for the contextual and superpixel parts, respectively. The method minimizes the energy using dynamic programming (DP) to estimate the semantic segmentation. We have implemented the algorithm [163] into the Automatic Labeling Environment (ALE), with the personal advice of the authors [106]. As with the original paper [163], we set the number of bag-of-words clusters to 512 and trained ALE using the CH1 dataset [163].

An Edge-Less Approach to Horizon Line Detection [5]. This approach also uses machine learning and DP to extract the horizon line from an image. The method assigns a classification score to each pixel, expressing the likelihood that the pixel belongs to the horizon line. As suggested by the authors, we used the SVM classifier trained by their training set [5]. Assuming that the horizon line extends from left to right (not top to bottom), the horizon line is extracted using DP, maximizing the sum of classification scores.

Fully Convolutional Networks (FCN) [120]. FCN achieve state-of-the-art results in semantic segmentation. For the given input image, the fully convolutional network produces a correspondingly-sized semantic segmentation image. We experimented with several semantic segmentation models (FCN-Xs) and selected the FCN-8s (three-stream, 8-pixel prediction stride), which gave us the best results, for further evaluation. We used a model trained for the 21-class (including background) PASCAL VOC segmentation task and finetuned for sky-foreground segmentation using the CH1 dataset [14].

4.5.4 Localization Performance

We evaluated the localization performance using several scenarios. In order to illustrate the performance of our implementation of the baseline method, we measured the performance on the original *CH1* dataset using the original *CH1* horizon lines (*CH1* area, *CH1* data, 4.8(b)). Since the *CH2* dataset does not contain horizon lines segmentations, we measured the performance of the *CH2* dataset using query horizon lines obtained by automatic segmentation of query images using three different methods (*CH2* area, *CH2* data, 4.9(a)). To study the difficulty of the *CH1*, *CH2*, and *GeoPose3K* datasets, we also evaluated the method on the *CH1* and *CH2* areas using *GeoPose3K* data (*CH1* area, *GeoPose3K* data, 4.8(c); *CH2* area, *GeoPose3K* data, 4.9(b), respectively). We also studied the method's total performance using the *GeoPose3K* images inside the largest *GP1* area, 4.9(c).



(a) localization area (b) *CH1* area, *CH1* data (c) *CH1* area, *GeoPose3K*

Figure 4.8: (a) trained localization areas: ***CH1* dataset area**, **CH2 dataset area** , **our *GP1* area** (largest); results of horizon-based localization on (b) *CH1* dataset: red—2.5° features, blue—10° features, green—combination of both 2.5° and 10° features; (c) *GeoPose3K* data in the *CH1* dataset area (yellow rectangle) using three automatic segmentation techniques—ALE (green), FCN (blue), Edge-Less (red).

CH1 area, CH1 data (Fig. 4.8(b)). We evaluated the performance of our implementation on the original *CH1* dataset [163] and used a database of horizons inside the yellow rectangle (Fig. 4.8(a)) and 203 query images from the *CH1* dataset. Like the authors of the baseline method [163], we visualized performance for 2.5° features, 10° features, and a combination of both. The obtained performance is a bit worse than in the original publication. We hypothesize that the main reason is in the data we use—the original method uses non-free DEM from the Federal Office of Topography swisstopo[6], which contains one sample per 2 m^2. We use publicly available DEM from viewfinderpanoramas [7], which contains one sample per 576 m^2.

CH1 area, *GeoPose3K* data (Fig. 4.8(c)). This experiment used 865 query images from *GeoPose3K* located inside the *CH1* area—yellow box (Fig. 4.8(a)). The fraction of correctly localized images is lower than in Fig. 4.8(b). This result might be caused by the lower accuracy of the horizon extraction algorithms (ALE, FCN-8s, Edge-Less) compared to those used in the original *CH1* dataset (ALE, guided by user). The *CH2* area with *CH2* and *GeoPose3K* data (Fig. 4.9(a) and 4.9(b)) also supports this assumption; the results look similar in both cases.

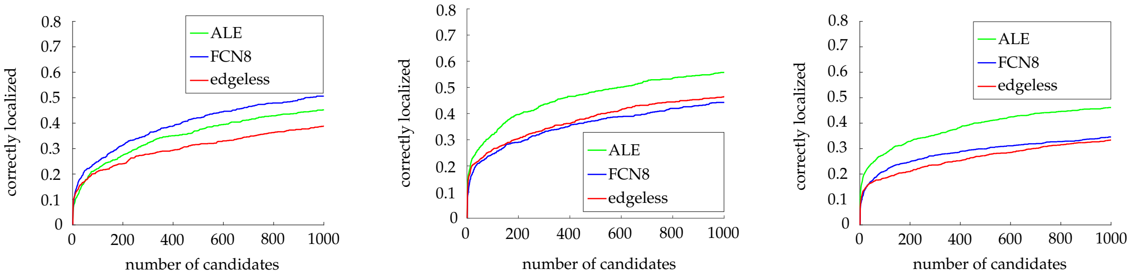
CH2 area, CH2 data (Fig. 4.9(a)). We used 949 query images from the *CH2* dataset located inside the *CH2* area—blue box in Fig. 4.8(a). The performance of this experiment can be directly compared to the *CH2* area—*GeoPose3K* experiment (Fig. 4.9(b)), since query horizon lines for both sets were extracted by the same (automatic) techniques. According to the results, the method performed a little bit worse on the *CH2* dataset.

CH2 area, GeoPose3K data (Fig. 4.9(b)). For this experiment, we used 791 images from the *GeoPose3K* dataset, located inside the *CH2* area—blue box in Fig. 4.8(a). The results agree with other experiments—the performance of the segmentation techniques is consistent with other experiments. ALE seems to be the best method for horizon line segmentation, FCN, and the EdgeLess approach scored similarly.

GP1 area, *GeoPose3K* data (Fig. 4.9(c)). The *GeoPose3K* dataset covers almost the whole Alps (Fig. 4.4). However, training such a large area for the horizon-based localization was not feasible due to hardware limitations. For this reason, we trained the *GP1* area (red area on (Fig. 4.8(a)), which is the largest area used for horizon-based localization so far (86,000 km^2). In this area, we evaluated the method using a subset of 1,151 images from *GeoPose3K*, which fit into the *GP1* area. The results of this experiment are in Fig. 4.9(c). The performance is slightly worse than in previous experiments (*CH1*, *CH2* areas, *GeoPose3K* data). We expected the performance drop since the geo-localization area is more than twice the size of the *CH1* and *CH2* areas. From this result, it seems that the geo-localization performance of the horizon line-based localization method [163] decreases only marginally with the increasing size of the geo-localization area.

[6] https://www.swisstopo.admin.ch

[7] http://www.viewfinderpanoramas.org



(a) *CH2* area, *CH2* data

(b) *CH2* area, *GeoPose3K*

(c) *GP1* area, *GeoPose3K*

Figure 4.9: results of horizon-based localization using three automatic segmentation algorithms—ALE (green), FCN (blue), Edge-Less (red); (a) *CH2* dataset using automatic segmentation; (b) *CH2* dataset area, using *GeoPose3K* images, (c) largest *GP1* area using *GeoPose3K* images.

4.5.5 Orientation Performance

Since *GeoPose3K* also contains *camera orientation* for each image, we evaluated the estimated heading (Fig. 4.10). To our knowledge, this is the first evaluation of heading estimated by the method of Saurer *et al.* [163]. We measured the difference between the ground truth heading and the estimated heading for each correct candidate. From Fig. 4.10, we can see that the orientation error peaks around 0°, and errors larger than several degrees are negligible. This result supports our expectations: for a correct location, the algorithm finds a correct heading estimate up to a small error of several degrees. We present a more in-depth analysis of the heading estimation error in Table 4.2. We measured the heading error on all three geolocalization areas – CH1, CH2, and GP1, using GeoPose3K data and calculated mean, standard deviation, and quantiles at 5% and 95%. The statistics correspond with Fig. 4.10. In the CH1 area, the Edge-less segmentation method achieved the best result; however, this is not consistent across other areas. CH2 and GP1 area likely contain more challenging images since the standard deviation is worse on CH2 than on CH1. In the largest GP1 area, the FCN8-s segmentation method has the lowest heading error according to the reported mean and standard deviation. On average, the lowest mean error in heading accuracy was achieved by ALE, which has other average statistics slightly deviated from the lowest observed values; however, the difference is not significantly different from other methods.

4.5.6 Experiments Summary

This section provided experimental results of the state-of-the-art horizon-based visual localization technique by Saurer *et al.* [163]. We evaluated both localization and heading

CH1 area, GeoPose3K, 865 images					
method	**mean**	**std**	**median**	**q = 0.95**	**q = 0.05**
Edge-less	**7.89**	**22.39**	**-0.76**	**6.62**	-8.78
FCN8-s	9.92	26.73	-1.66	15.74	**-8.77**
ALE	16.10	36.36	-0.80	80.36	-25.54
CH2 area, GeoPose3K, 791 images					
Edge-less	36.28	51.88	**0.39**	124.48	-120.18
FCN8-s	25.68	43.62	-1.80	107.84	-122.17
ALE	**11.76**	**32.04**	-0.79	**29.23**	**-9.08**
GP1 area, GeoPose3K, 1151 images					
Edge-less	14.76	34.89	**-0.18**	105.22	**-7.35**
FCN8-s	**13.00**	**33.97**	-1.24	106.21	-7.53
ALE	18.26	36.10	-0.61	**98.61**	-20.38
Average					
Edge-less	19.64	36.39	**-0.18**	78.77	-45.44
FCN8-s	16.20	**34.77**	-1.57	76.60	-46.16
ALE	**15.37**	34.83	-0.73	**69.40**	**-18.33**

Table 4.2: Statistics of the camera orientation error in degrees for a localization experiment on GeoPose3K data using three automatic segmentation techniques. Symbol $q = 0.95$ denotes quantile at 0.95.

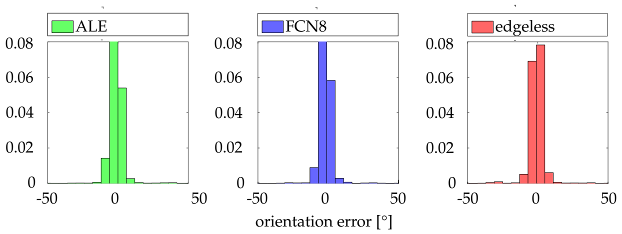


Figure 4.10: Normalized histograms of camera orientation error (in degrees) for localization experiment on *CH1* area and *GeoPose3K* dataset using three automatic segmentation techniques—ALE (green), FCN (blue), Edge-Less (red).

estimation performance. For evaluation, we used the original CH1 and CH2 datasets and compared the achieved performance with our GeoPose3K dataset. We also conducted the largest horizon-based localization performance experiment ever, with the use of the GeoPose3K dataset and a GP1 area of 86,000 km^2. We identified a large performance gap between automatically estimated horizon lines and manually corrected ones provided with the CH1 dataset. Usually, the method could localize around 15% of top-1 candiates with a localization error below 1 km using our dataset. The performance was two times better with the original CH1 dataset: the method localized around 30% of top-1 candidates with a localization error below 1 km. The best method for automatic horizon line extraction is, according to our experiments, Automatic Labeling Environment (ALE) [106] (which scored best in 3 out of 4 experiments), and the second is the Fully Connected Networks (FCN-8s) approach (which scored best in 1 out of 4 experiments).

For the first time, we evaluated the heading estimation performance of the horizon-based localization method by Saurer *et al.* [163]. Through our experiment, we illustrated that candidates located up to a distance of 1 km from the ground truth showed a heading error of around a few degrees, and more significant discrepancies from the ground truth heading are rare. In other words, a correctly localized image also implies a correctly estimated heading. However, such an estimated heading is only an approximate estimation, since the mean error varies between 7.89° and 36.28° across various scenarios.

4.6 Future Applications

GeoPose3K dataset is a rich source of information for solving geo-localization, camera orientation, and other computer vision and image processing problems. Besides geographic location and the camera orientation, it contains additional synthetic data to train, evaluate, and compare existing and future algorithms. Let us briefly summarize possible future applications of this dataset.

Depth Estimation from a Single Image. Depth is an essential cue for image processing algorithms, like single image haze removal [79, 51]. Depth estimation from a single image is an ill-posed problem since there is no unique mapping from a single RGB image to RGB-D. We must take prior and contextual information into account in order to obtain feasible estimates. The prior is usually chosen arbitrarily, such as a dark channel [79]. However, we can train the prior or the whole end-to-end estimation process, given our synthetic depth and normals.

Sun Position from Illumination. Sun position is a viable feature for location recognition [107]. Previous work estimates the sun's position given a set of temporal images. We might train the sun's position estimation from a single image using our synthetic illumination data in an end-to-end manner in future work.

Semantic Segmentation. Semantic segments proved to be usable for camera orientation assessment [12, A3]. In Chapter 6, we used semantic labels from the *GeoPose3K* dataset to fine-tune a semantic segmentation method to estimate semantic segmentation similar to the rendered ones.

4.7 Chapter Summary

In this chapter, we presented the *GeoPose3K* dataset. We showed that the dataset is unique and valuable for the training and evaluation of methods in the context of visual *geo-localization* and *camera pose* estimation. We demonstrated an approach to semi-automatic dataset acquisition using an improved camera orientation estimation algorithm. We performed an in-depth analysis of dataset properties and provided the most extensive baseline evaluation on a geo-localization task using a state-of-the-art visual geo-localization algorithm. Our experiments demonstrated that the *GeoPose3K* is usable for camera orientation and geo-localization evaluation, and the difficulty is on par with original *CH1* [14] and *CH2* [163] datasets. Additionally, we proposed several unconventional future applications which the dataset enables us to develop.

Chapter 5

Building Large-scale Cross-domain Datasets Using SfM

Structure from motion (SfM) techniques for 3D reconstruction and camera pose estimation are frequently used to recover 3D geometry of the scene [42, 113, 70, 95, 115]. The datasets reconstructed using SfM contain camera parameters relative to each other and are defined up to the rotation and scale. The recent computer vision trend uses large amounts of images for which the camera parameters were estimated using SfM and use them to learn solutions for various problems directly from data. Problems which were addressed with the help of large scale datasets reconstructed using SfM include camera pose estimation [95, 26], depth estimation [115] or keypoint detection and description [126, 48]. This chapter presents our research towards a precise alignment of unsorted photography collections with the terrain model, which allowed us to build large datasets of photographs and their precisely aligned counterparts synthetically rendered from the terrain model.

Contributions. We present two approaches to align unsorted collections of photographs with a terrain model using SfM technique. We show, that although SfM reconstruction is challenging in outdoor environments, it can be used to build extensive datasets of images precisely aligned with the terrain model. The first approach uses SfM to reconstruct a 3D scene represented as a point cloud which is subsequently aligned with the terrain using GPS tags from the input photographs. The initial alignment is further refined by precise registration of the 3D point cloud with the terrain model. Second, we propose a novel approach to Structure-from-Motion using terrain reference which reconstructs the 3D scene jointly from photographs and rendered images. The rendered images, for which we know exact camera parameters, help fixing the reconstruction to avoid drift. Using both methods, we automatically reconstructed several scenes containing more than 20,000 photographs precisely aligned with the terrain.

5.1 Related Work

The goal of our work is to align unstructured collections of real photographs with a rendered terrain. However, the existing SfM approaches [185, 1, 218, 217, 130, 80, 170] alone cannot recover the scene in absolute world coordinates, which would allow the alignment with existing terrain models, *e.g.*, geo-tagged DEMs. Wang *et al.* [208] used the SfM technique in a two-step process to first reconstruct a 3D scene from photographs and then align it with the terrain. To align the reconstructed 3D scene with a city model, the authors use GPS positions of cameras for initial geo-localization. In the next step, they apply a rigid fine-tuning of the scene with 3D building models using Iterative Closest Points (ICP). As they use vanishing points to estimate the reconstructed scene up vector, the method is limited to urban scenes with linear features. In contrast, we need a method independent of linear features that are usually not available in outdoor mountain sceneries. This chapter introduces two different approaches to overcome this problem and shows that we can use SfM methods in the cross-domain scenario to align a photograph with the rendered terrain model.

Section 5.2 introduces our first approach that leverages a two-step reconstruct-then-align approach. Similarly to Wang *et al.* [208], we use SfM to reconstruct the scene, which we subsequently align with the terrain model [A4]. In contrast to Wang *et al.* [208], our method does not estimate the up vector using vanishing points. We demonstrate that with proper processing, Flickr images' noisy GPS positions provide sufficiently precise initial geo-registration to enable further refinement with the terrain model.

The second approach, introduced in Section 5.3, completely removes the alignment step—we directly match photographs with the terrain model to reconstruct a scene with absolute coordinates [A5]. To our knowledge, our method is the first to propose a 3D SfM reconstruction jointly using real photographs and rendered imagery to achieve an implicit geo-registration.

5.2 Scene Reconstruction and Alignment

To reconstruct a dataset, we collect images from a specific area of interest. We obtain ground-level images through the Flickr API, querying for the specific geo-extent covering the area of interest, which we additionally restrict to a specific time interval. Restricting the time interval ensures that the downloaded photographs are taken during roughly the same season, improving matching and reconstruction by eliminating seasonal changes.

Some of the images retrieved with a location filter may contain irrelevant data instead of natural outdoor scenes (*e.g.*, indoor images, pets, close-ups of vegetation, or portraits of hikers). We filter them to improve the efficiency of our algorithm. To select only relevant images, we apply a scene understanding neural network (ResNet18) trained on the Places365 dataset [231] to find images that are most likely both *outdoor* and *natural*.

Given an input image, the network estimates matching scores for a list of semantic categories defined in the Places365 database. The semantic category is a high-level representation of a place, e.g., *bedroom*, *beach*, or *mountain*. For each semantic category, the Places365 dataset defines whether it is indoor or outdoor. Per-image, we select the semantic categories with the ten highest scores; if most of them are indoor, the image is classified as indoor and otherwise outdoor.

To implement the natural/unnatural classification, we use the image attributes from the SUN attribute dataset [136]. Semantically overlapping image attributes describe scenes with fine granularity. We cluster the attributes as either natural (non-urban images) or unnatural (everything else). Examples of natural attributes include *foliage*, *leaves*, or *hiking*; examples of unnatural attributes are *pavement*, *carpet*, or *stressful*. The CNN estimates per-attribute correlations for an input image. We sum all correlations for natural attributes and subtract correlations for the unnatural attributes. If the outcome is greater than zero, then we classify the image as natural.

5.2.1 Scene Reconstruction

We tested several publicly available Structure from Motion pipelines [218, 217, 130, 185, 170] for scene reconstruction. We obtained the best results using the publicly available COLMAP implementation [170]. We found important to use approximate matching with a vocabulary tree and an enhanced voting strategy for fast spatial verification [168] since exhaustive matching is significantly slower. We use a 256k vocabulary tree provided by the COLMAP authors[1]. The typical reconstruction time of a dataset of 4k photographs was several hours on a desktop PC with NVIDIA 970 GTX GPU.

For geo-registration using GPS from Flickr images, we use a robust least median of squares (LMeds) combined with RANSAC [229] using the Euclidean distance of the reconstructed camera position and the corresponding GPS position (residual). Instead of minimizing the sum of squared residuals, we minimize their median, which is more robust towards outliers. Using this minimization approach, we estimate a similarity transformation to transform (translate, rotate, and scale) the scene into world coordinates.

5.2.2 Fine-tuning

Because of uncertainties in camera configuration, GPS location, and other parameters, there is no guarantee that the initial geo-registration matches the known terrain. To remedy this, we refine the initial geo-registration by minimizing the Euclidean distance between the reconstructed 3D point cloud and the DEM terrain data. We segment the point

[1] https://demuc.de/colmap/

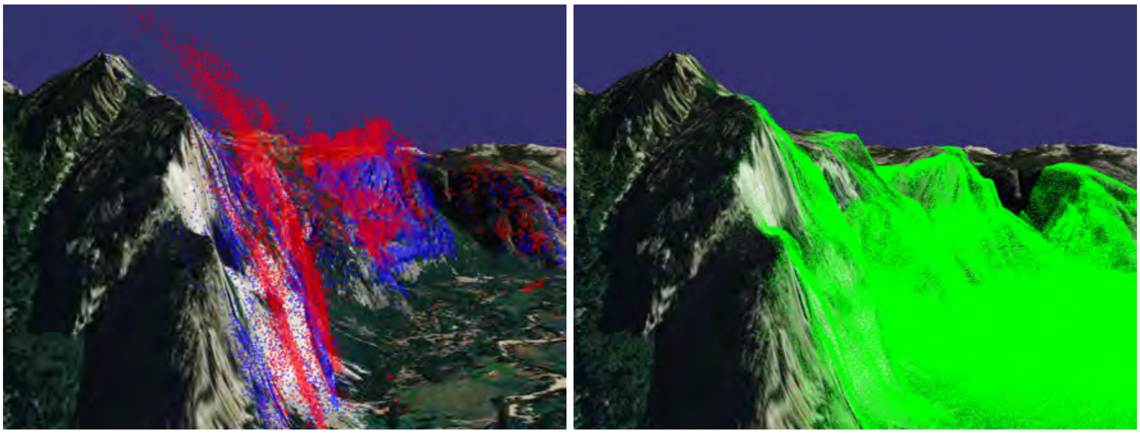


Figure 5.1: Alignment of input (red) point cloud with the reference (green) point cloud sampled from the terrain using Iterative Closest Points. The blue point cloud is the result. Map data © Mapbox, © OpenStreetMap.

cloud into disjoint clusters so that two points in the same cluster are at most 1 km apart from each other. We calculate a bounding box for each cluster and sample the terrain on a grid with 10 m spacing. We align the reconstructed 3D point cloud and the sampled terrain using ICP with the Libpointmatcher library [141] with default parameters. The algorithm first reduces the input and reference point clouds (see Figure 5.1) by random sampling, keeping 75% of all points. Next, the algorithm iteratively performs a series of steps:

1. Each point is matched to its nearest neighbors in the Euclidean space.
2. Points too far from the reference point cloud (outliers) are removed (85% of points with the smallest distance are kept).
3. Minimization of point-to-plane distance is performed [220].
4. Check if convergence or the maximum number of iterations (40) has been reached.

After registering the model, we are often left with mismatches between the photo content and the virtual terrain, mostly due to wrong information about camera configuration (*e.g.*, focal length, or exact GPS position). Furthermore, because of the DEM's limited sampling rate, some cameras may end up below the virtual terrain after the ICP alignment, which we solve by moving them vertically to the terrain height. However, both of these problems introduce errors in camera orientation parameters.

To correct the registration errors, we leverage our knowledge of the correspondences between 2D points o_i observed in the photographs and the point cloud 3D points p_i aligned with the virtual terrain. We use these correspondences to optimize the orientation parameters using the Kabsch Algorithm [88]. We project the 2D observations o_i using camera

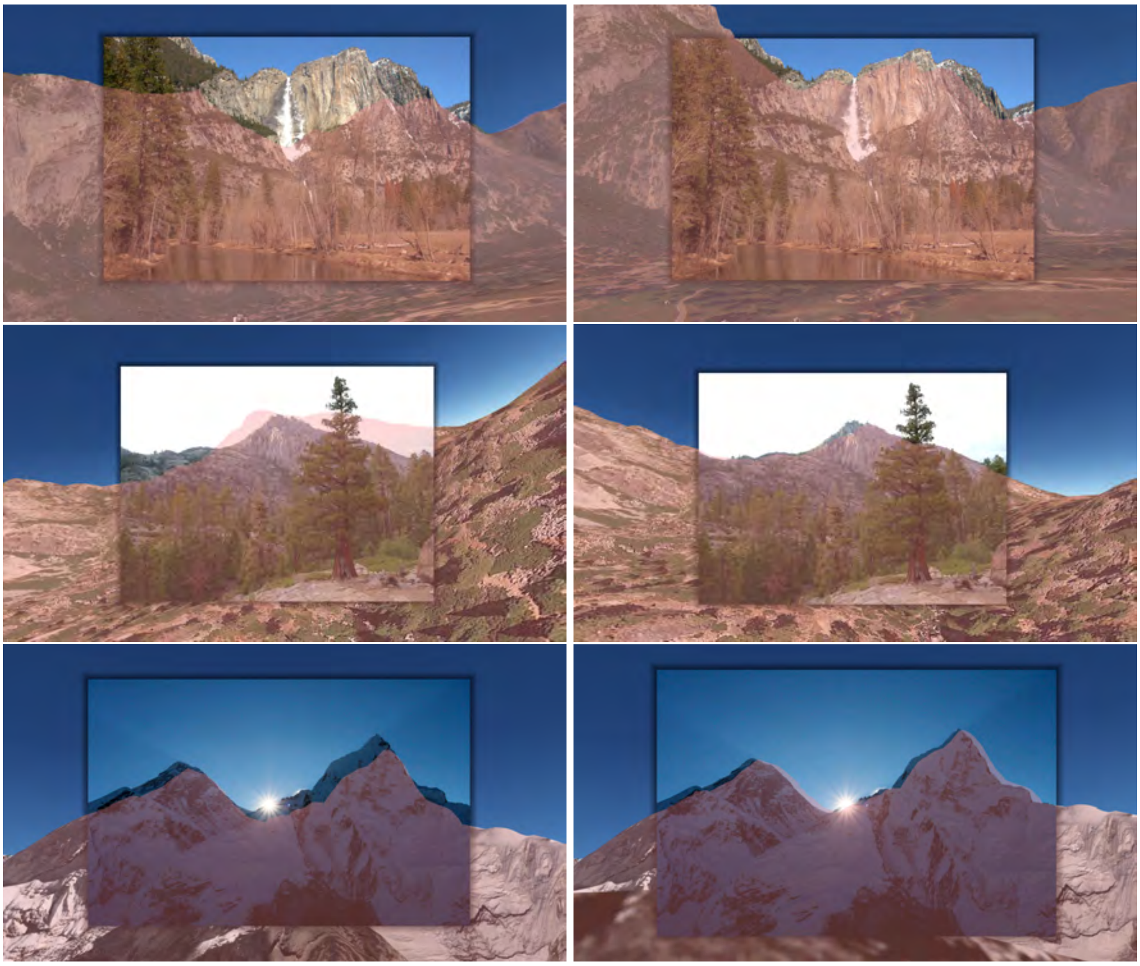


Figure 5.2: Examples of images before (left) and after point-cloud to terrain alignment using ICP (right). Top row: Yosemite Waterfall, CA, USA, middle row: Jakes Peak at the Lake Tahoe, CA, USA, bottom row: Mount Everest, Nepal. Map data © 2018 Google.

dataset	I_m	I_{mr}	Me(e_p) [m]	Me(e_p)-ICP [m]	μr_e [px]
Nepal	2401	901	1624.62	819.98	0.41
Tahoe	302	78	2814.24	72.82	0.88
Tatras	4146	297	2908.59	2410.21	0.47
Yosemite	4173	2094	14041.70	348.33	0.50
Matterhorn	33829	9018	836.89	440.48	0.51

Table 5.1: Number of input images before reconstruction (I_m), number of reconstructed images (I_{mr}), median alignment error (Me(e_p)) of the point cloud and the terrain before and after ICP, and mean RMSE of the reprojection (μr_e). The median alignment error Me(e_p) is significantly lower after alignment using ICP.

parameters into 3D points $\overline{p_i}$ based on the Euclidean distance between the camera center and the corresponding 3D point p_i from the point cloud. We subtract centroids from both sets and calculate the rotation matrix using the Kabsch algorithm $\mathbf{R} = K(\overline{p_i}, p_i)$. We show the results of the fine-tuning in Figure 5.2. The implementation of our method is publicly available[2].

Using this approach, we reconstructed five datasets from five different locations across the globe. We captured the *Nepal* dataset at the Himalaya mountains in Nepal; the *Tahoe* dataset comes from the Lake Tahoe in California, USA; the *Tatras* dataset consists of photographs from the High Tatra mountains in Slovakia; the *Yosemite* dataset comes from the Yosemite National Park in California, USA; and the *Matterhorn* dataset comes from the European Alps. Table 5.1 illustrates the matching accuracy of the reconstructed 3D point cloud with the sampled terrain. Because a reconstructed model usually contains a small number of outliers, we report the median euclidean distance $\text{Me}(e_p)$ between each 3D point p_i from the point cloud and its closest point from the sampled terrain $n(p_i)$:

$$\text{Me}(e_p) = \text{Me}\left\{\forall i \in \{1, \ldots, N\} : e_p[p_i, n(p_i)]\right\}, \tag{5.1}$$

where e_p has been defined in Eq. 3.16, and Me denotes the median.

To illustrate the reconstruction's accuracy, we also include the mean of the reprojection root mean squared error (RMSE) μr_e across all cameras in the given dataset:

$$\mu r_e = \frac{1}{N_c} \sum_{i=0}^{N_c - 1} \sqrt{\frac{1}{N_p(P_i)} \sum_{j=0}^{N_p(P_i)-1} \|\Gamma(p_j, P_i) - o_j\|_2^2}, \tag{5.2}$$

where N_c is the number of cameras in the dataset, $N_p(P_i)$ is the number of points observed by camera pose P_i, $\Gamma(p_j, P_i)$ is the projection function which projects 3D point p_j into the image plane using camera pose P_i, and o_j is the 2D observation corresponding to the 3D point p_j.

5.3 Direct Cross-domain Reconstruction

Existing methods reconstruct a sparse 3D model from photographs [218, 217, 130, 185, 170] and then align it with the terrain model [208] using point cloud alignment methods. These methods generally work well for areas with a dense coverage of ground-level photographs. However, sometimes the coverage density is too low, and we need an alternative approach which makes the reconstruction more robust and stable.

To this end, we propose a registration method that aligns photographs via a DEM-guided structure-from-motion, in which the known camera parameters and geometry of

[2] https://github.com/brejchajan/itr

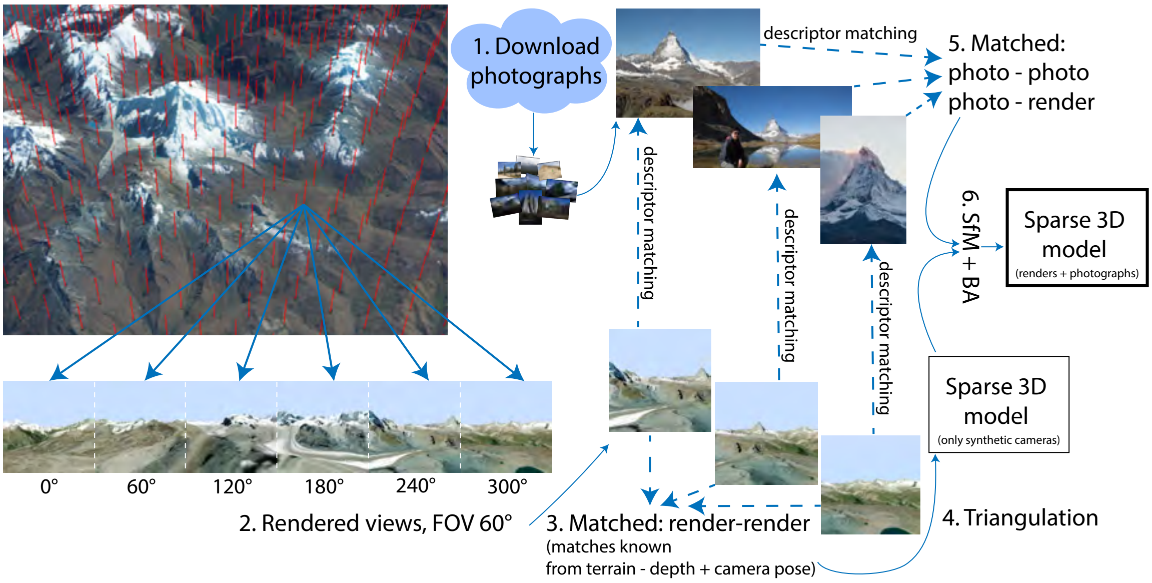


Figure 5.3: **Structure-from-motion with a terrain reference for automatic cross-domain dataset generation.** In the area of interest, camera positions are sampled on a regular grid (red markers). At each position, 6 views covering the full panorama are rendered. A sparse 3D model is created from the synthetic data using known camera poses and scene geometry. Each photograph is then localized to the synthetic sparse 3D model. Image credit, photographs left to right: John Bohlmeyer (https://flic.kr/p/gm3zRQ), Tony Tsang (https://flic.kr/p/gWmPbU), distantranges (https://flic.kr/p/gJCPui).

the DEM domain help overcome ambiguous matches and lack of data in the photo domain. As input, we use photographs from an online service (Fig. 5.3-1.), such as Flickr.com, and download all photos within a given radius. The radius may vary between 10 km to 30 km. For the same area, we also render panoramic images sampled 1 km apart on a regular grid (Fig. 5.3-2.). Specifically, we experimented with six areas from the European Alps, and with one area from the South American Andes. We illustrate the number of rendered images I_r for each area in Tab. 5.2. We render six images with 60° field-of-view, each rotated by 60° around the vertical axis for each sampled position. For each rendered image, we store a depth map, full camera pose, and detected keypoints and descriptors using D2Net [48]. For rendered images, we calculate matches directly from the terrain geometry using the stored camera poses and depth maps—no descriptor matching between rendered images is needed (Fig. 5.3-3.). We reproject each keypoint from a rendered pair of images to 3D based on the rendered camera pose and the depth of the keypoint from the depth map. We find keypoint correspondences by searching for the nearest neighbor, far away at most M meters. With the DEM resolution of 30 m/pixel, we used $M = 40$ m. We can then obtain an initial sparse 3D model by triangulating the respective matches (Fig. 5.3-4.).

dataset	I_r	I_m	I_{mr}	μr_e [px]
Alps Eiger	3072	12280	2281	1.32
Alps Grande Casse	2700	12849	1347	1.29
Alps Gran Paradiso	2700	3728	592	1.28
Alps Chamonix	4926	6248	1908	1.34
Alps Ortler	2700	10436	2103	1.29
Alps Wildspitze	2700	9882	1011	1.27
Andes Huascaran	5664	3008	184	1.31

Table 5.2: Number of input renders (I_r), number of input images before reconstruction (I_m), number of reconstructed images (I_{mr}), and mean RMSE of the reprojection (μr_e).

In the next step, we extract keypoints and descriptors from the input photographs using D2Net. We match the input photographs to every other photograph *and* synthetic render using descriptor matching (Fig. 5.3-5.) and localize them to the terrain model using SfM (Fig. 5.3-6.). We use global bundle adjustment to refine camera parameters belonging to photographs and 3D points, while the rendered cameras have fixed all parameters since they are known precisely.

Notably, while existing single-domain feature descriptors are not robust to the photo-DEM domain gap, we can overcome this limitation by the sheer volume of synthetic data. Most of the matches will be within the same domain (e.g., photo to photo), and only a small handful need to successfully match to DEM images for the entire photo domain model to be accurately registered. Finally, we check each reconstructed photograph's location with the terrain model and prune photographs below, or more than 100 m above the terrain since they are unlikely to be localized precisely.

This approach successfully geo-registered photographs in every tested area. Importantly, this approach was successful even for areas with low density of photographs—in Gran Paradiso (Alps) and in Huascaran (Andes) areas we had only between 3-4 photographs per square kilometer. In total, we localized 9,426 photographs using this approach; our implementation and reconstructed datasets are publicly available[3]. We present the numbers of reconstructed images per dataset and mean reprojection error in Tab. 5.2. We also show some qualitative results in Fig. 5.4. According to visual inspection, the alignment is reasonably precise and consistent across the majority of the results. Our method can localize and align images captured at challenging lighting conditions, such as night photographs (Fig. 5.4, middle-right), or horizon line occlusions (Fig. 5.4, bottom line). Fail-

[3] https://github.com/brejchajan/LandscapeAR

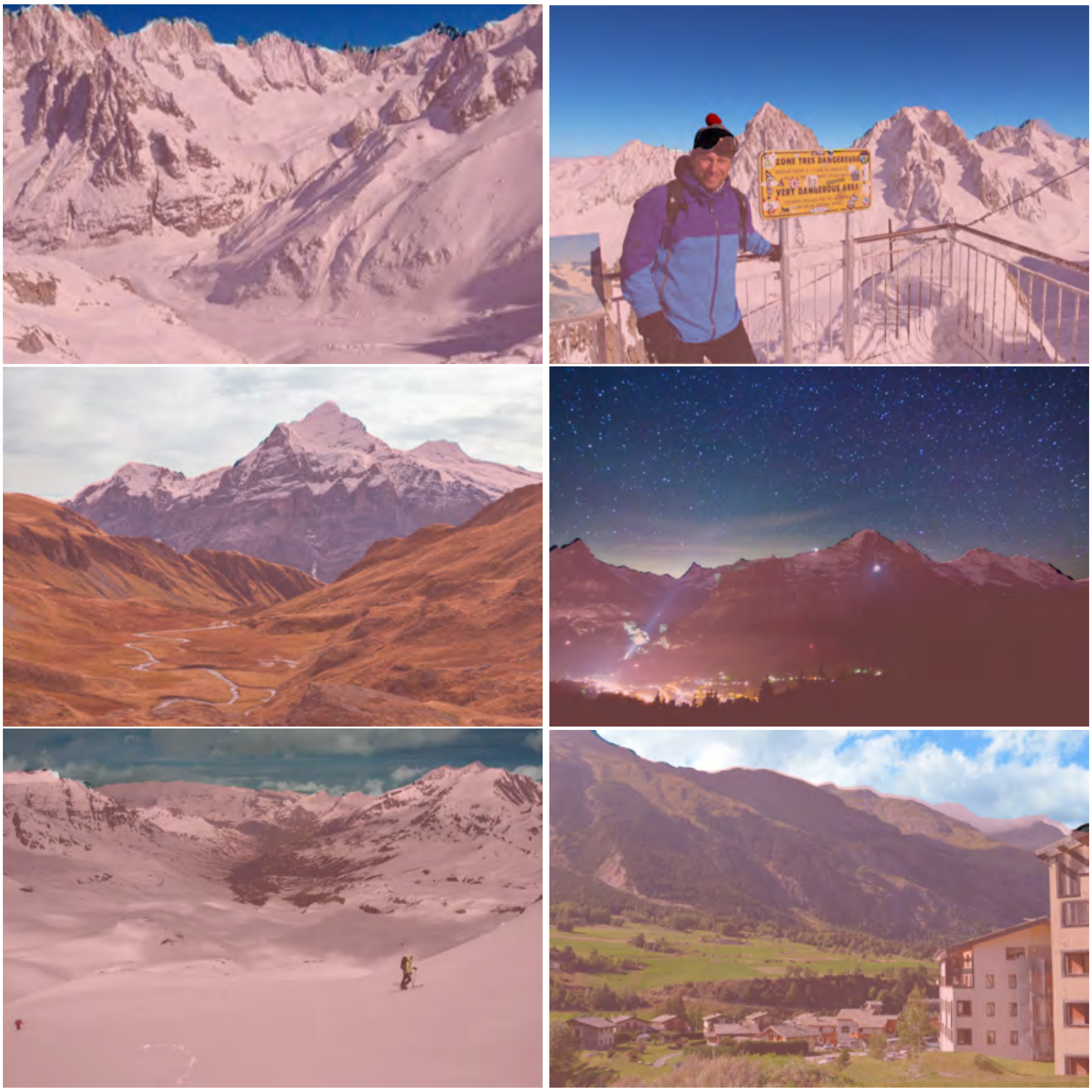


Figure 5.4: Examples of alignment generated by finding camera pose using SfM with terrain reference. The synthetic terrain rendered from DEM is overlaid over the photographs and highlighted with red color. Top row: Chamonix, European Alps, image credit (left to right): Kenneth Berger (https://flic.kr/p/khHgkc, Owen Richard (https://flic.kr/p/krFqWM); middle row: Eiger, Europen Alps, image credit (left to right): distantranges (https://flic.kr/p/gJBZZv), Tom Fear (https://flic.kr/p/gLPF1w); bottom row: Grande Casse, European Alps, image credit (left to right): antoine.pardigon (https://flic.kr/p/nKV3hh), Jean-Marie Zanoni (https://flic.kr/p/p18rDc).

ure cases are sparse and are usually caused by the insufficient resolution of the rendered DEM foreground.

5.4 Chapter Summary

We presented two *SfM-based* approaches for the acquisition of photographs precisely aligned with a terrain model. First, we introduced our modification of the reconstruct-then-align approach, initially used in urban areas [208]. We showed that using GPS information from downloaded internet photographs is enough to align the reconstructed scene with the terrain model in outdoor environments [A4]. In contrast to the original approach [208], our method does not need to detect vanishing points, which is difficult due to the lack of straight segments in outdoor sceneries. However, many outdoor scenes are almost impossible to reconstruct using photographs solely due to the following issues. Some outdoor areas are covered by internet imagery only sparsely, and the scene's appearance varies significantly across individual photographs. Furthermore, internet imagery's internal camera parameters are usually unknown and vary significantly due to a broad range of consumer devices. Because of these shortcomings, the SfM reconstruction often drifts, effectively disallowing subsequent alignment with the terrain model.

We solved this by introducing a novel method, which aligns outdoor photographs with a sparse 3D terrain model implicitly during the reconstruction [A5]. Our novel approach uses real photographs and synthetic renders of the terrain with fixed camera parameters. We estimate camera parameters for photographs during the reconstruction by matching the photographs with each other and with the renders. Photo-to-photo and photo-to-render matching allow us to constrain the reconstruction with the known camera parameters (coupled with synthetically rendered images) and optimize parameters of the real cameras (coupled with the photographs) and parts of the scene depicted on the photographs, but missing from the terrain model. Using both methods, we automatically reconstructed several datasets and presented quantitative and qualitative results of the image-to-terrain alignment.

Part III

Visual Localization by Photo-to-terrain Matching and its Application

In this part, we propose two novel methods for calibrating extrinsic camera parameters. The first method uses the combination of features based on edges and semantic segmentation to match a query image with a synthetic panorama rendered at a known position. The second method studies the full camera pose estimation by proposing a novel cross-domain descriptor for matching keypoints between the query photograph and a rendered terrain model covered with a satellite texture. Finally, we propose a novel application that uses visual localization to present photographs in an immersive virtual environment. The application allows users to easily re-visit the places from their vacation or perform a photographs' showcase in a virtual environment to familiarize others with a novel yet unvisited location.

Chapter 6

Camera Orientation Estimation in Natural Scenes Using Semantic Cues

A variety of works approached the camera orientation estimation [146, 133, 15, 100, 12, 143, 142, 39]. With the knowledge of camera orientation and position in the world, we can infer answers to questions such as: “Is it possible to move forward?” or “What are we looking at?” While state-of-the-art data-driven methods [95, 7] can answer such questions, they are focused mainly on urban areas. In contrast, this chapter focuses on camera orientation estimation in mountainous areas. Knowledge of camera orientation may be valuable for scene understanding and organizing large databases of photographs. Furthermore, camera orientation may augment other sensors in robots, UAVs, or helicopters for automatic navigation. Several works on camera orientation estimation in mountainous areas were developed [15, 12, 142, 52, 132]. However, the problem remains challenging for real-world images, as illustrated by our experiments.

While the position of a photograph is often recorded with the GPS sensor, personal photographs and internet images often lack information about the camera orientation. The knowledge of accurate camera orientation opens up interesting applications and facilitates difficult image recognition tasks. For example, images with known camera pose can be augmented with information from geospatial databases and used in augmented and virtual reality applications. Existing solutions to camera orientation estimation in mountainous scenes rely on matching a query image with a terrain model [15, 12, 142]. In general, these methods are based on aligning query image features (edge maps) with synthetic edges generated from the terrain model. While we also use a terrain model as a reference, we do not rely solely on the edge information. In contrast to previous works, we align areal features that complement edge information. Specifically, the development of semantic segmentation methods allows us to employ matching based on semantic segments. We map terrain features, such as forests, bodies of water, and glaciers from a geospatial (GIS) database to a digital elevation model (DEM) and render into a panorama image containing semantic segments (Fig. 6.1(a)). From the query image, we extract semantic segments (Fig. 6.1(b)) using recent semantic segmentation methods [106, 120, 36]. We match the

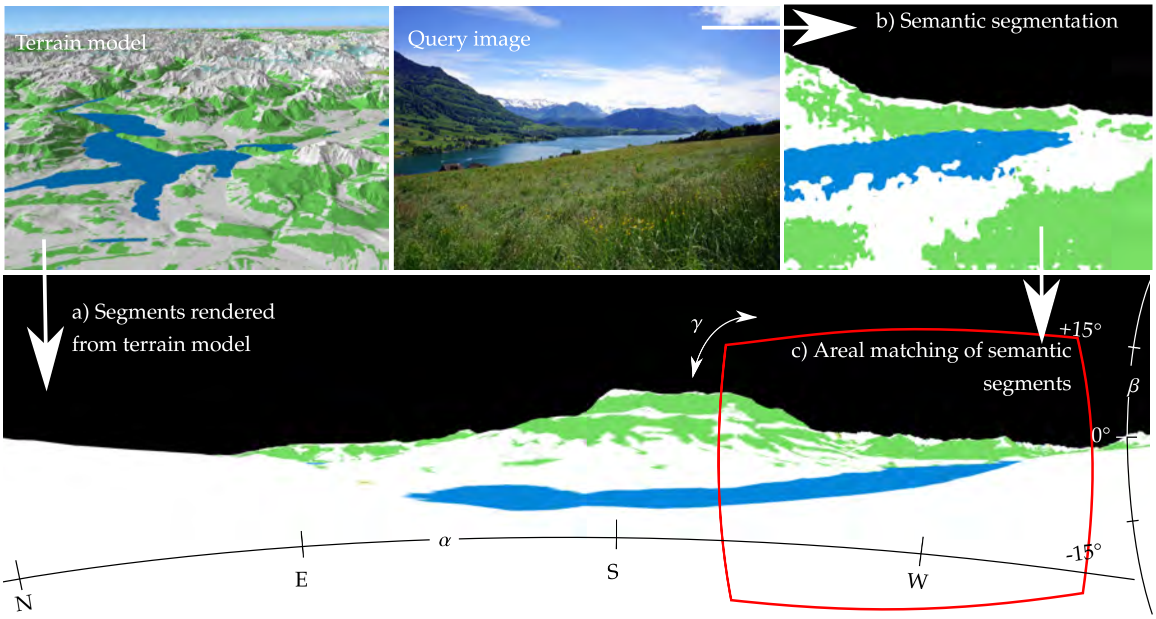

Terrain model
Query image
b) Semantic segmentation
a) Segments rendered from terrain model
γ
c) Areal matching of semantic segments
+15°
β
0°
-15°
α
N
E
S
W

Figure 6.1: Overview of the proposed method. (a) Synthetic semantic segments are rendered using terrain model and geospatial database. (b) Query image is segmented via semantic segmentation method. (c) Semantic segments from query image are aligned with synthetic semantic segments and camera orientation (α,β,γ) is recovered.

query and the panorama (Fig. 6.1(c)) to estimate camera orientation. We estimate a correspondence between the query image and the synthetic panorama based on the similarity of the same class's semantic segments. Intuitively, spatial relationships between different semantic classes disambiguate in-plane rotations. To exploit these spatial relationships, we introduce confidence fusion (**CF**), which prefers camera orientations with the highest confidence agreement across all semantic classes. The benefit of the proposed technique is the possibility to naturally fuse confidence estimates of different modalities, such as different segment classes and edge maps.

Contributions. We propose a novel method for aligning a single image to a digital terrain model. To our knowledge, we are the first to consider a joint combination of semantic segments and edges to match an image with a rendered panorama of the terrain. We train semantic segmentation on a synthetically rendered dataset and show that synthetic data is needed to achieve reasonable accuracies when used for orientation estimation in a mountainous environment. To enable matching of several semantic segment classes and an edge map with the rendered panorama, we propose a novel confidence fusion (**CF**) method that fuses individual beliefs to achieve better accuracy. Our experiments show that the proposed method outperforms state of the art on publicly available test sets—GeoPose3K [A1], Venturi Mountain dataset [142], and CH1 dataset [163].

6.1 Related Work

Works dealing with natural scenes have shown that the horizon line is a distinctive and relatively stable feature for camera orientation and position estimation [69, 200, 37, 163]. However, relying solely on the horizon line can be misleading, since there are many situations when the horizon line is ill-defined, non-descriptive, or completely invisible:

1. View from an elevated location to a flat landscape implies a flat horizon line.

2. Foreground objects, like trees, often contaminate the horizon line.

3. The horizon line is not visible due to camera pitch (images without the sky).

Recent works dealing with camera orientation estimation with a fixed position for outdoor and mountainous scenes are based on a query image's alignment with a terrain model [22, 15, 145, 12, 142]. Instead of using a single horizon line, Baboud *et al.* [15] and Porzi *et al.* [142] used edge maps to align a query image to a synthetically rendered terrain silhouettes. In this chapter we show that it is beneficial to combine edge features with other modalities, such as low-frequency semantic segments, which complement the high-frequency edges.

Most closely to our semantic segmentation-based approach presented in this chapter, Baatz *et al.* [12] used semantic segments for the image alignment. They extracted binary descriptors capturing the spatial relationships between different classes of segments. However, the descriptors encode local changes between neighboring segments, meaning that this technique exploits only segment boundaries. The boundaries are usually inaccurate for real-world cases, rendering the method unstable. We address this issue by proposing a method for areal matching of semantic segments. The main idea is that segment areas should match well, unlike potentially imprecise segment boundaries.

Several approaches for camera position and orientation estimation based on semantic segments were also developed for urban environments. Senlet *et al.* [172] and Castaldo *et al.* [32] used semantic segments for matching an input image with a GIS map to estimate a camera position, but their approach cannot recover the camera orientation precisely. Armagan *et al.* [9] proposed an iterative approach to fine-tune camera position and orientation based on semantic segmentation with known camera position and orientation estimate. In contrast to their work, our approach is more general as it does not need *any* initial camera orientation estimate.

6.2 Orientation Estimation Using Semantic Cues

We aim to estimate camera orientation using a digital terrain model for a given query image. Similarly to Baboud *et al.* [15] the basic idea is to project the query image onto the sphere and align it with the spherical panorama rendered from the model. The correct alignment then defines the searched camera orientation. We assume that the position $\hat{C} = (\phi, \lambda, h)$ parametrized by latitude, longitude, and elevation, and the horizontal field-of-view θ of the query image I are known. The goal is to find a rotation $\hat{R} \in \mathrm{SO}(3)$ of the camera frame relative to the frame of the digital terrain. We render the terrain model with synthetic semantic segments as a spherical $360° \times 180°$ panorama (see Fig. 6.1(a)), with $\hat{C}$ as the unit sphere center. A projective query image containing estimated semantic segments is projected on the unit sphere as well. The query image is scaled to cover the part of the unit sphere corresponding to its field-of-view by a factor $s = \frac{\theta}{2\pi I_w}$, where I_w is the query image's width.

6.2.1 Cross-correlation as a Measure of Confidence

To estimate the camera orientation $\hat{R} = (\alpha, \beta, \gamma)$, we compute a matching confidence $c(\alpha, \beta, \gamma)$ over all possible combinations of rotations $\alpha \in \langle 0°, 360° \rangle$, $\beta \in \langle 0°, 180° \rangle$, $\gamma \in \langle 0°, 360° \rangle$ (see Fig. 6.1(a) for respective rotations). We also define a confidence $c_k > 0$ for semantic segment class k and later fuse all confidences into the total confidence c. The combination of parameters maximizing the total confidence defines the camera orientation estimate $\hat{R} = \arg\max_{\alpha,\beta,\gamma}(c(\alpha, \beta, \gamma))$.

We propose the confidence c_k to be a cross-correlation of the query and panorama on SO(3), containing semantic segments of class k. Similarly to Baboud *et al.* [15], we exploit the cross-correlation theorem for efficient computation of cross-correlation in the Fourier domain. Cross-correlation of two real-valued functions f and p on SO(3) is similar to ordinary 2D cross-correlation, but we are integrating over a sphere (S^2):

$$\forall \hat{R} \in \mathrm{SO}(3) : f \star p(\hat{R}) = \int_{S^2} f(\omega) p(\hat{R}^{-1}\omega) d\omega. \tag{6.1}$$

For each class k, we construct two spherical functions p_k (query segments) and f_k (synthetic segments) as follows. To obtain strictly positive confidence, we need the spherical functions to be strictly positive as well. We sample both query segments and synthetic segments of class k on a unit sphere, where we assign one to pixels containing the segment of class k and $\epsilon \to 0^+$ to pixels that contain other segment classes, where ϵ is a small positive constant. However, calculating cross-correlation for a single segment class k using p_k and f_k may not be sufficient for correct alignment (see the top line in Fig. 6.2). In this case, the cross-correlation is maximized for all rotations, where $p_k(\hat{R}) \leq f_k$. This way, segments

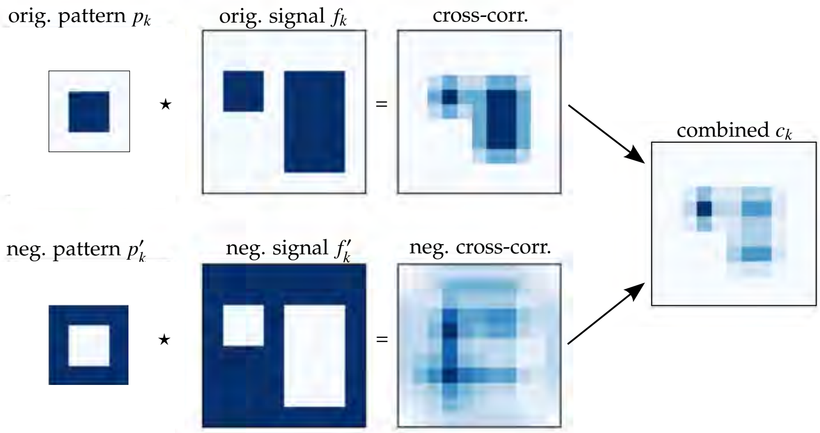


Figure 6.2: Illustration of the cross-correlation behavior for two functions $p_k > 0$ and $f_k > 0$, which are, without loss of generality, defined on $\mathbb{R}^2$ for this example. White color denotes $\epsilon \to 0^+$, darker color denotes a higher value. In the first line, the cross-correlation is maximized even for translations, where surroundings of the pattern are not in agreement with the signal. The inverted pattern and signal on the second line create a complementary cross-correlation map. When the two cross-correlations are combined, the maximum value is correctly in place where both the pattern and its surroundings overlap the largest areas.

from the query image tend to "hide" inside larger synthetic segments of the panorama image. In other words, there are large areas with the maximum cross-correlation value. To alleviate this problem, we divide the computation of class confidence c_k into two steps, as illustrated in Fig. 6.2. The first step is the cross-correlation $\forall \hat{R} \in \mathrm{SO}(3) : f_k \star p_k(\hat{R})$, given the class k. The second step is a complementary cross-correlation with inverted spherical functions $f'_k = 1 + \epsilon - f_k$, $p'_k = 1 + \epsilon - p_k$. The combined cross-correlation, which equals to class confidence c_k across all rotations $\hat{R} \in \mathrm{SO}(3)$ is then calculated as:

$$\forall \hat{R} \in \mathrm{SO}(3) : c_k(\hat{R}) = (f_k \star p_k(\hat{R}))(f'_k \star p_k(\hat{R})'). \tag{6.2}$$

Intuitively, the first cross-correlation maximizes rotations where query segments overlap the synthetic segments, while the second cross-correlation maximizes rotations where the surroundings of query segments overlap the surroundings of the synthetic segments. By multiplying the two cross-correlation results, we robustly enforce rotations where the overlap of both the segment area and its surroundings is maximized.

Please note that the two-step cross-correlation is necessary and cannot be replaced by +1 and -1 encoding for the segment and the background, respectively. Consider the situation in Fig. 6.3, where we compare our two-step correlation to a single-step version. The leftmost pixel matches background, the second pixel matches the foreground, and two pixels on the right do not match (background on the foreground). Since two of four pixels

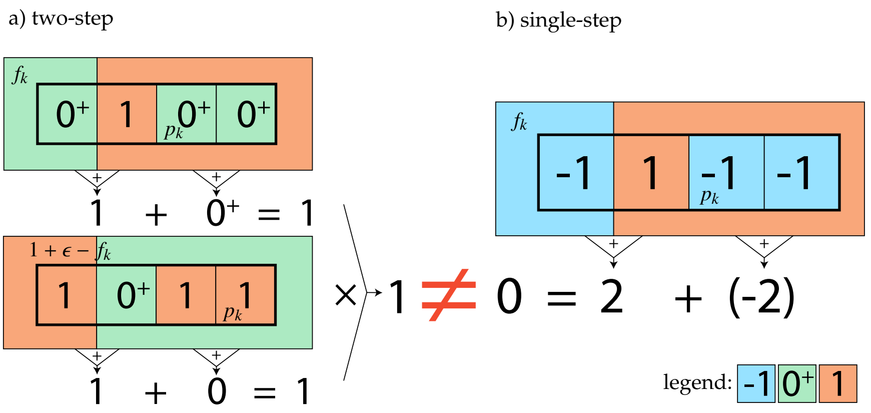


Figure 6.3: Two-pass cross-correlation is not equivalent to a single-pass using negative values. Our two-pass approach calculates number of correct pixels and disregards the wrong pixels if *both* the foreground and the background are matched. In contrast, the single-pass penalizes the wrong pixels, which leads to result incompatible with our definition of confidence.

match the foreground or background, we expect the confidence to be greater than ϵ. Our two-step approach maximizes the correct overlap of segments (and returns 1) while the single-step method is biased by non-matching regions (and returns 0).

6.2.2 Confidence Fusion

So far, we have considered the confidence of a single segment class. A single segment class k is usually not sufficiently descriptive to constrain the correct rotation since the semantic segment areas are often similar for many rotations. Mutual spatial relationships between different segment classes help to disambiguate the correct rotation. While a single segment class does not disambiguate the roll angle (see Fig. 6.4), the combination of two segments gives a single precise maximum, located at the desired rotation (see Fig. 6.4, combined *c*).

With the assumption that the segments are correctly detected in the query image, and no segments are missing from the rendered panorama, the highest confidence across *all* fused classes would determine the correct rotation. To calculate it, we would simply calculate the product of confidences across all classes:

$$\forall \hat{R} \in \mathrm{SO}(3) : c(\hat{R}) = \prod_k (c_k(\hat{R})). \tag{6.3}$$

However, the assumption of correct detection and complete model cannot be fully satisfied in real-world applications. In this case, the wrongly detected segment could cause drift from the correct solution. To compensate mistakes in the detection or missing parts in the model, we propose to compute the Confidence Fusion (**CF** framework) as a weighted

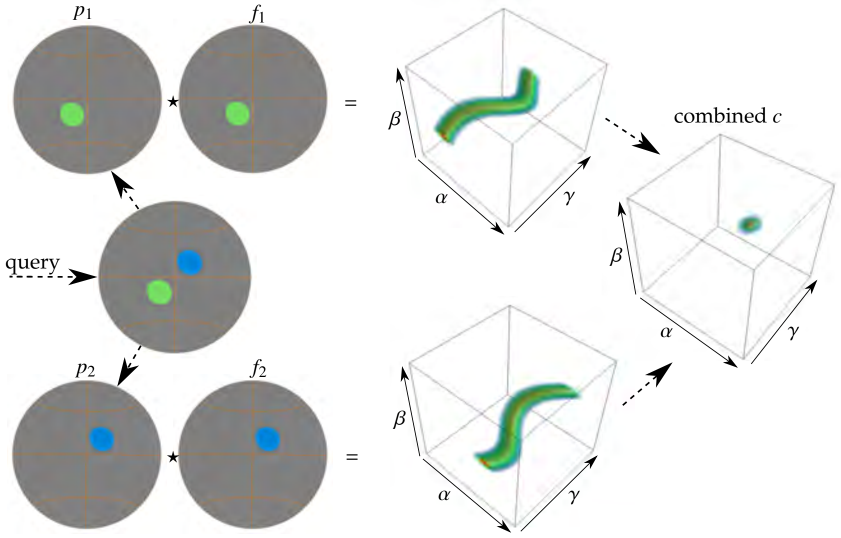


Figure 6.4: Synthetic experiment illustrating the confidence fusion. Cross-correlations are visualized as a heatmap over orientations (α, β, γ), which form a cube. The query image contains two circles, each circle represents one semantic segment (classes of the segments are different). In this case, cross-correlation of a single segment class does not disambiguate the roll angle (γ). On contrary, the fusion of confidence maps maximizes at a single orientation, as visualized in the rightmost cube.

geometric mean:

$$\forall \hat{R} \in \mathrm{SO}(3) : c(\hat{R}) = \prod_k (c_k(\hat{R}))^{w_k}. \tag{6.4}$$

The importance of the segment class k can now be tuned by the weight $w_k \in \langle 0, 1 \rangle$: the weight should be small for wrongly detected segment classes and high for classes that are detected and rendered correctly.

6.2.3 Weight Estimation for Confidence Fusion

We can estimate the weights in many ways. We tried to regress them directly based on the GeoPose3K training set, but this approach has not proved to be robust across different datasets. We can borrow the robust estimation of the weights for a fusion of multiple densities from Ajgl and Šimandl [6] (Theorem 2), where the authors derive the computation of weights in the sense of minimization of maximal Kullback-Leibler divergence between the fused confidence c and the class confidences c_k. The method needs to be used carefully in order to keep the computational complexity reasonably low and to allow suppression of wrongly detected class confidences. In theory, we want to estimate the weights $w_k \in w$:

$$w = \underset{\substack{\omega_k : 0 \le \omega_k \le 1, \\ \sum_k \omega_k = 1}}{\arg\max} \; -\ln k(\omega), \tag{6.5}$$

where

$$k(\omega) = \int_{\mathrm{SO}(3)} \prod_k (c_k(\hat{R}))^{\omega_k} d\hat{R}. \tag{6.6}$$

The $\hat{R} \in \mathrm{SO}(3)$ defines a rotation on 3D rotation group, $w_k \in w$ is the estimated weight of a segment class k, and $\omega_k \in \omega$ represents the space of all possible weights for a segment class k. This method finds the "best average" between the segment class confidences. However, according to our experience, this one-shot fusion does not assign low values to wrongly detected class confidences, and its computational complexity rises exponentially with the number of classes. According to our observation, segment classes with smaller segment areas tend to be imprecise more often than classes with larger segment areas. The matched area of a segment class k directly corresponds to the integral $\mathcal{S}_k$ of the class confidence c_k over all rotations $\hat{R}$: $\mathcal{S}_k = \int_{\mathrm{SO}(3)} c_k(\hat{R}) d\hat{R}$. If the segment class k matches a smaller area, $\mathcal{S}_k$ is smaller; if the segment class k matches a larger area, $\mathcal{S}_k$ is larger. We use this property to suppress the confidences with lower integral and to reduce computational complexity. We fuse the confidences with the iterative pairwise fusion. This approach provides the best results in our application, but in general, it is suboptimal in terms of Kullback-Leibler divergence [6]. We sort the class confidences c_k according to their integral $\mathcal{S}_k$ starting with the lowest one. We begin the fusion with $c(\hat{R})$ being a uniform distribution over all possible rotations, and calculate updated $c(\hat{R})$ by fusing it with class confidence c_k one at a time:

$$\forall \hat{R} \in \mathrm{SO}(3) : c(\hat{R}) = \frac{1}{k(w_k)} c_k(\hat{R})^{w_k} c(\hat{R})^{1-w_k}, \tag{6.7}$$

$$k(\omega) = \int_{\mathrm{SO}(3)} \frac{1}{k(\omega)} c_k(\hat{R})^{\omega} c(\hat{R})^{1-\omega} d\hat{R}, \tag{6.8}$$

where the weight w_k for current segment k is calculated as

$$w_k = \arg\min_{\omega} k(\omega). \tag{6.9}$$

We repeat this process for each class's confidence c_k. The impact of class confidence c_k is lower for confidences fused earlier, thus reducing the classes' impact with smaller segment area.

However, we can avoid the expensive calculation of weights altogether. We observed that class confidences c_k are potentially incorrect for segments covering small areas, as small segments may be wrongly detected or occluded. We solved this problem by setting the weights empirically. If the area covered by the segment class k in the query or panorama image is lower than a threshold t[1], we simply turn off the segment class k by setting its value $w_k = 0$. For the remaining segment classes, we set $w_k = 1$. This simple ap-

[1] We use $t = 0.1\%$ of the total image area (found experimentally).

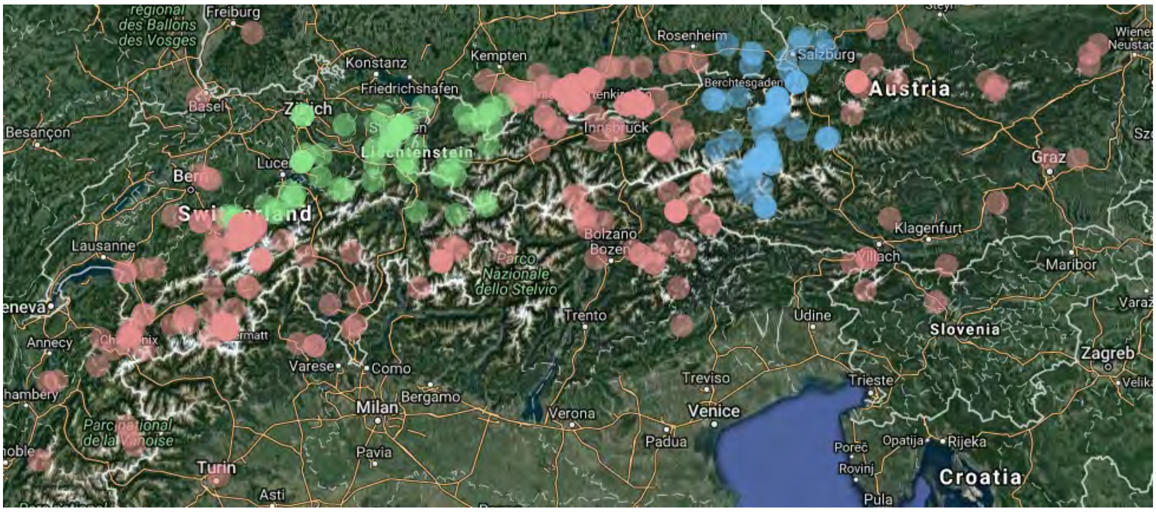


Figure 6.5: Geographical distribution of GeoPose3K train, validation and test sets. Map source credit: Google Maps.

proach significantly outperforms non-weighted fusion (eq. 6.3) and provides comparable results to the approach of Ajgl and Šimandl [6] in our application.

Semantic segments and edge features. Our Confidence Fusion framework (**CF**, eq. 6.4) can use any nonnegative result based on spherical cross-correlation; it is not limited to semantic segments only. Most methods employ edge features to match a real image with a terrain model [15, 14, 163, 200, 37, 53]. Our goal is to show that it is highly beneficial to combine edge features with other cues, such as the semantic segments. We use edge detector trained to estimate silhouette edges similar to the rendered ones [A1] (see Sec. 4.2.2). To calculate confidence based on edge features, we use a cross-correlation metric developed exclusively for edges, VCC-2011 [15], for which we replace negative values with $\epsilon \to 0^+$.

6.2.4 Semantic Segmentation

To match a query image with rendered semantic segments, we need a segmentation method to estimate semantic segments that are visually similar to the rendered counterparts. To achieve this, we fine-tune several state-of-the-art semantic segmentation models. Please note that the fine-tuning using a synthetic dataset is a crucial step in the whole **CF** framework, and it is one of the contributions of this work.

We consider two state-of-the-art CNN architectures: FCN [120] and Deeplab-v2-VGG-16 [36], and one non-CNN method which we use as a reference: Automatic Labeling Environment (ALE) [106]. We start with SiftFlow and Pascal-Context models for FCN8s, and similarly, for training DeepLab-v2, we use VGG-16 as an initial model. We fine-tuned all models on the GeoPose3K dataset [A1], which contains synthetic semantic labels for more than 3,000 images registered into the 3D terrain model. We split GeoPose3K into the train (1927 images), validation (472 images), and test sets (516 images), so that these three sets are geographically disjoint, see Fig. 6.5. This way, we ensure there are no similar images across the train, validation, and test sets. We optimize the geographical distribution of images, so the sets contain a similar amount of semantic segments per class (measured in pixels). The definition of our train/validation/test splits is available on our project webpage[2].

The GeoPose3K dataset contains, in total, 14 classes for semantic segmentation, including the sky. Unfortunately, many segment classes, such as sinkhole or bare-rock are available only for a limited subset of images. Segments of these classes often span a small area of the image, which reduces their descriptivity. Motivated by this observation, we selected the following subset of semantic segment classes, which cover a sufficient number of images: *mountain*, *sky*, *forest*, *water bodies*, and *glacier*. To fine-tune these classes using FCN8s and DeepLab-v2, we replaced the last classification neural network layer with a layer containing our own five classes.

6.3 Experiments

In this section, we provide an in-depth evaluation of the proposed camera orientation estimation. We use three publicly available data sets—GeoPose3K test set (516 test photos), CH1 dataset (203 photos) [163], and Venturi Mountain Dataset [142] (12 videos). The original CH1 dataset [163] does not contain camera orientation ground truths. However, the GeoPose3K contains images from the CH1 dataset and provides camera orientation ground truth [A1] (see Sec. 4.2). We held out the GeoPose3K, the CH1 test set, and the Venturi dataset from semantic segmentation training and used it only for testing. The presented evaluation is the most extensive analysis of camera orientation estimation methods in a natural environment without device sensors (compass, accelerometer, gyroscope). We compare our work directly with Baboud *et al.* [15], Porzi *et al.* [142], and our implementation of Saurer *et al.* [163]. Since Baatz *et al.* [12] and previous methods [146, 133, 145] use a limited number of private images for their evaluation, we could not directy compare our approach with these methods.

Evaluation metric. To compare our approach with the recent work of Porzi *et al.* [142], we use the *orientation error* measure as defined in Eq. 3.17 in Chapter 3. We calculate and plot a cumulative distribution of the orientation error, where fractions of images have the orientation error equal to or lower than the given threshold. A random baseline illustrates

[2] http://cphoto.fit.vutbr.cz/semantic-orientation/

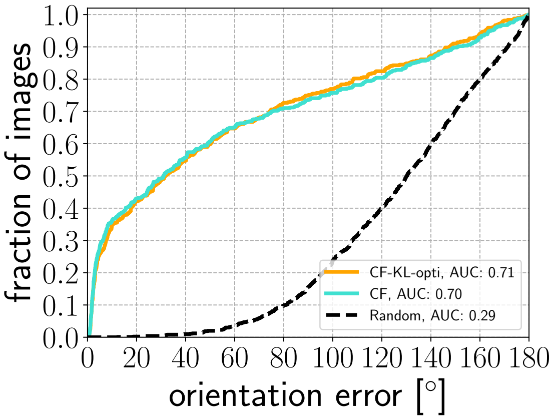


Figure 6.6: Comparison of the performance between the framework using iterative pairwise fusion based on KL divergence **CF-KL-opti** (see Eq. 6.7), and our approximate solution based on empirically found weights **CF**.

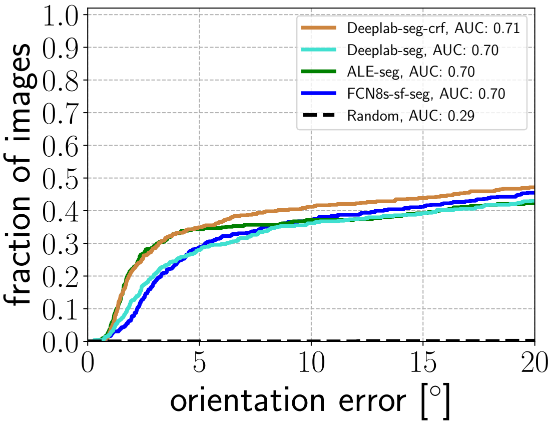


Figure 6.7: Performance of **CF** framework with different segmentation methods. The best—Deeplab with CRF (AUC: 0.71); other methods scored similarly (AUC: 0.70).

what the probability of guessing an orientation is. For better clarity, we also give a measure of Area Under Curve (AUC), where AUC = 1 is, in theory, the best possible result.

6.3.1 Weight Estimation Assessment

First, we compare two approaches to finding weights for the Confidence Fusion framework. In Fig. 6.6, we plot the results of semantic segment fusion **CF-KL-opti** using weights found by iterative pairwise fusion based on KL divergence (see Eq. 6.7) measure as proposed by Ajgl and Šimandl [6]. However, as this approach is relatively slow, we developed an approximate solution described in Sec. 6.2.3, denoted as **CF** in Fig. 6.6. On the one hand, both approaches give relatively similar results, and our approximation does not hurt the performance much. On the other hand, our approximation is much faster, and we use it in all the following experiments. Please note that Fig. 6.6 is a plot for the whole 180°search space, making it difficult to discern tiny differences close to zero. In the following experiments, we zoom-in to the range of $[0°, 20°]$ to ease readability.

6.3.2 Evaluation of Semantic Segmentation Methods

We select a semantic segmentation method for our orientation estimation framework using standard semantic segmentation metrics, namely *mean accuracy* and *mean Intersection over Union (mIU)*. These metrics, shown in Tab. 6.1, illustrate that both Deeplab with and without Conditional Random Fields (CRF), are the methods of choice. Since the metrics

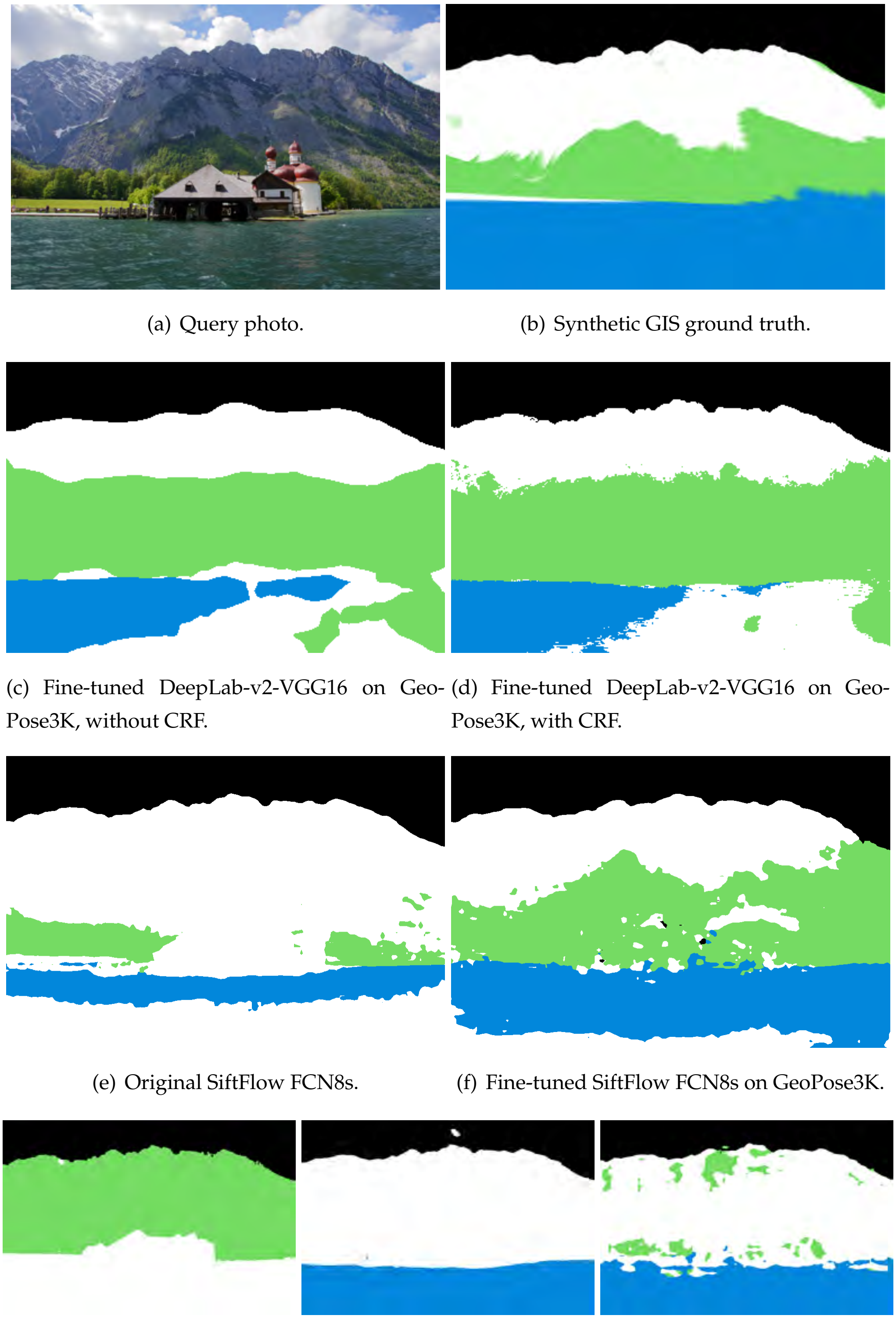
	DeepLab-v2 VGG16		**DeepLab-v2 VGG16 + CRF**		**FCN8s SiftFlow**		**FCN8s Pascal-Context**		**ALE**		**Naive baseline**	
mACC	**0.63**		0.62		0.59		0.54		0.61		0.20	
mIU	**0.53**		0.52		0.46		0.38		0.46		0.07	
	IU	**ACC**	**IU**	**ACC**	**IU**	**ACC**	**IU**	**ACC**	**IU**	**ACC**	**IU**	**ACC**
mountain	**0.60**	0.78	**0.60**	**0.79**	0.56	0.77	0.44	0.60	0.49	0.60	0.00	0.00
sky	**0.89**	**0.93**	**0.89**	**0.93**	**0.89**	**0.93**	0.82	0.89	0.79	0.91	0.35	1.00
forest	**0.38**	0.53	0.37	0.52	0.34	0.48	0.32	**0.57**	0.33	0.56	0.00	0.00
water	**0.44**	**0.55**	**0.44**	0.54	0.31	0.51	0.17	0.43	0.30	0.47	0.00	0.00
glacier	0.36	0.37	0.31	0.32	0.21	0.24	0.14	0.19	**0.40**	**0.49**	0.00	0.00

Table 6.1: Results of semantic segmentation methods trained with GeoPose3K. Results are measured on GeoPose3K test set; accuracy (*ACC*) and intersection over union (*IU*) are measured per class independently, mean pixel accuracy over all classes is denoted by *mACC*, and mean intersection over union over all classes is denoted as *mIU*. Last column represents a naive segmentation into a single class (sky), which has the largest *prior* probability in the GeoPose3K dataset.

are based on the ratio of correctly classified pixels, we expect that the best method based on these metrics is also the best for our camera orientation estimation framework. We verified this expectation by testing our orientation estimation framework with the semantic segmentation methods listed in Tab. 6.1. We show the results of this experiment in Fig. 6.7. Deeplab with CRF (Deeplab-seg-crf, AUC: 0.71) achieved the best result, but other segmentation methods—Deeplab without CRF, FCN8s, and ALE scored almost the same (AUC: 0.70). According to visual inspection (see Fig. 6.8), CNNs are slightly more successful in ignoring objects not present in the digital terrain model. We use DeepLab for all following experiments with our Confidence Fusion (**CF**) framework. In Fig. 6.8, we may also notice visual differences between original and fine-tuned models. We can see that the fine-tuned model Fig. 6.8(f) generates segmentations much more similar to the synthetic render Fig. 6.8(b) compared to the initial model Fig. 6.8(e).

6.3.3 The Impact of Cross-correlation Resolution

For calculating cross-correlation in SO(3) using Fourier transform (FFT), we use publicly available SOFT package [101]. The precision of the cross-correlation and computation time and memory footprint are driven by two factors—the input resolution of the spherical functions and the cross-correlation output resolution. Higher input resolution implies a more precise sampling of input spherical functions. The resolution of the output drives sampling of the resulting cross-correlation. Please, note that lower input and output resolutions do not restrict the search space to any particular orientation—we search a full 3D rotation, no matter what resolutions are selected.



(a) Query photo.

(b) Synthetic GIS ground truth.

(c) Fine-tuned DeepLab-v2-VGG16 on GeoPose3K, without CRF.

(d) Fine-tuned DeepLab-v2-VGG16 on GeoPose3K, with CRF.

(e) Original SiftFlow FCN8s.

(f) Fine-tuned SiftFlow FCN8s on GeoPose3K.

(g) ALE trained on GeoPose3K.

(h) Original Pascal-Context FCN8s.

(i) Fine-tuned Pascal-Context FCN8s on GeoPose3K.

Figure 6.8: Illustration of mountain scene semantic segmentation using DeepLab-v2 based on VGG-16 model, FCN8s, and ALE, before and after fine-tuning on the GeoPose3K dataset. Photo credit: Allie Caulfield https://flic.kr/p/9VryJg.

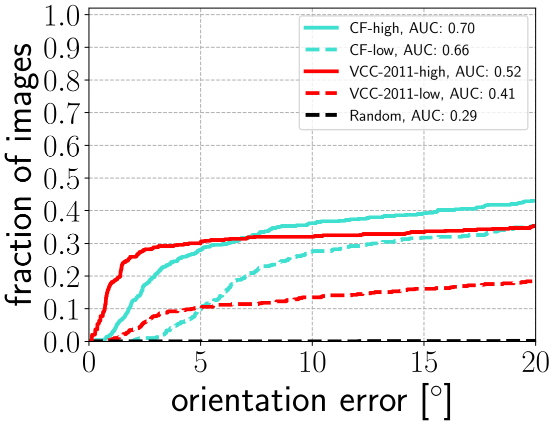


Figure 6.9: Comparison our **CF** framework using semantic segments with edge-based VCC-2011 [15] on high (solid curves) and low (dashed curves) resolutions.

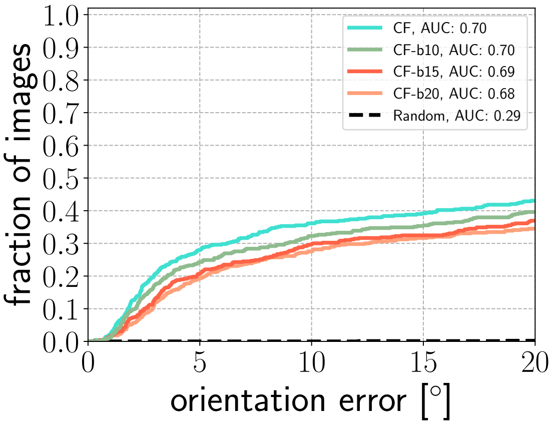


Figure 6.10: Original semantic segments were smoothed by gaussian blur using three different kernel radii 10 px (b10), 15 px (b15) and 20 px (b20).

In general, we expect that lower resolution (coarser sampling) of the functions would decrease the method's precision. Intuitively, coarser sampling might negatively affect high-frequency functions more than low-frequency functions. Semantic segments encode low-frequency information, while edge features encode mainly high frequencies. According to this observation, we expect that using lower input and output resolution affects the precision of cross-correlation of semantic segments much less than cross-correlation of edges.

To verify this hypothesis, we run an experiment to compare the effect of input and output resolutions on the achieved accuracy (see Fig. 6.9). We consider two versions of input and output resolution. The first version is a low resolution, with the input resolution of 1024 samples and the output resolution of 128 samples (see dashed curves in Fig. 6.9). The low resolution yields fast evaluation (about 1.5 seconds per cross-correlation), and the orientation estimation of a single query lasts at most 30 seconds (depending on the number of segment classes). However, the result confidence is stored in a cube of size $(128)^3$, yielding almost 3° per bin, which may increase the orientation error. The second version is a high resolution one, where we set the input resolution to 4096 and the output to 512 samples (see solid curves in Fig. 6.9).

The experiment confirmed our expectation that using a lower resolution for cross-correlating semantic segments does not dramatically increase the orientation error (see cyan solid, vs. cyan dashed curve in Fig. 6.9). A lower resolution extensively reduces the time and memory footprint (from 45 seconds per cross-correlation to just 1.5 seconds, and from 12GB of memory to just 247MB on high and low resolution, respectively). Compared to semantic segments, the edges contain higher frequencies, which are more affected by subsampling. In the case of edge-based cross-correlation (VCC-2011 [15]), the high-resolution variant brings a decent improvement in terms of accuracy over the low resolution (see Fig. 6.9 red solid vs. red dashed curve). This result is in agreement with our expectations as well. The relative indifference to subsampling is an advantage of using segments over the edges.

6.3.4 Importance of Segment Boundaries

To ensure that our approach factually does not boil down to matching boundary edges of semantic segments, we conducted an experiment in which we suppressed the importance of segment boundaries by gaussian blur. We blurred the original query and synthetic segments with three different kernel radii—10 px (0.43°), 15 px (0.65°) and 20 px (0.86°). The blur removes the hard boundaries of semantic segments and reduces their impact. Since segment areas' boundaries tend to be imprecise, we expect that suppressing their importance should not negatively affect the result. We show the achieved performance in Fig. 6.10. We achieved the best performance using a non-blurred and 10 px kernel radius (AUC: 0.70). For larger kernel radii the accuracy dropped only slightly, having AUC: 0.69 and 0.68 in the case of 15 px and 20 px radius, respectively. We see that the blur does not affect the results significantly, which illustrates that potentially inaccurate segment boundaries are not very informative for camera orientation estimation using our **CF** method. The main information resides in segment areas and rough shapes.

6.3.5 Are Edges and Semantic Areas Complementary?

The previous experiment suggests that segment areas encode the primary information, unlike the segment boundaries. Intuitively, segment areas correspond to low-frequency information, while edge features encode high frequencies. This property should allow *combining* both types of features to increase orientation estimation accuracy. We calculate two confidences: one using VCC-2011, and the second one using semantic segments. We obtain the final result by fusion of both confidences with our **CF** framework. Since VCC-2011 penalizes query and silhouette edge crossings, the result of VCC-2011 may contain negative values. To use the VCC-2011 result in our **CF** framework, we clamp negative values with $\epsilon \rightarrow 0^+$ before fusion.

The following experiment confirmed our expectation that the combination of edges and semantic segments improves the orientation estimation result. We used the GeoPose3K test set to measure the orientation error of VCC-2011 [15] (Fig. 6.11—red curve, AUC: 0.52). The cyan curve in Fig. 6.11 denotes the result obtained by our **CF** frame-

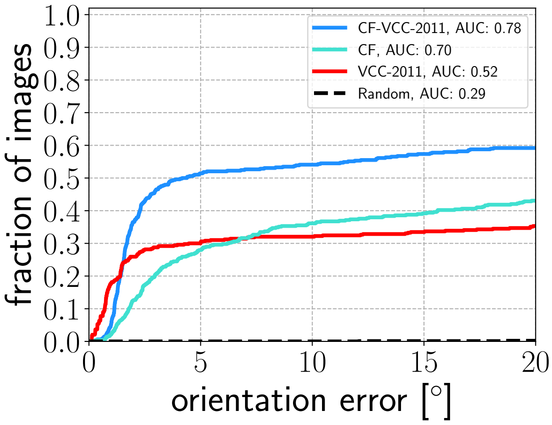


Figure 6.11: Comparison of the edge-based VCC-2011 [15], our **CF** framework using semantic segments, and combination of both approaches. We use our **CF** framework to fuse semantic segments and edges (**CF-VCC-2011**), which gives the best result.

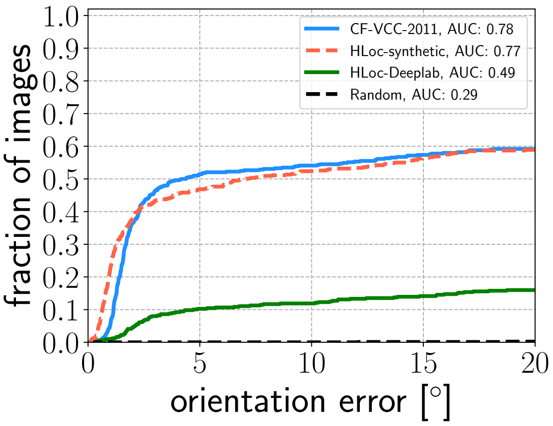


Figure 6.12: Our **CF** compared to **HLoc** [163] on GeoPose3K test set. **CF** (blue) using automatic segmentation and **HLoc** using synthetic sky segmentation (dashed) perform similarly; **HLoc** using automatic sky segmentation (green) performs worse.

work using semantic segments only—AUC: 0.70. We can see that our method using semantic segments yields better performance than edge-based VCC-2011. Using our **CF** framework, the combination of both (VCC-2011 and segments) scored the best performance (Fig. 6.11—blue curve, AUC: 0.78). The difference between using edges and semantic segments is 18%. Furthermore, the combined result brings an improvement of 26% over the VCC-2011. We recorded similar results on the CH1 dataset [163], and the Venturi Mountain dataset [142] see Fig. 6.14 and Tab. B.1, CF-VCC-2011 vs. VCC-2011 vs. CF. We conclude that according to this experiment, the semantic and edge features are complementary. Combining both approaches improves the camera orientation performance significantly.

6.3.6 Comparison with State-of-the-art

This section presents a series of experiments showing that our **CF** framework produces more accurate results than existing state-of-the-art methods. With the authors' personal advice, we have reimplemented a horizon line-based localization method (abbreviated as *HLoc*[3]) by Saurer *et al.* [163] into the same DEM rendering pipeline as **CF** and evaluated its ability to find correct camera orientation with known camera position. We used the best dir&loc [163] scheme to calculate a heading estimate of a given query and a panorama horizon line, followed by ICP, to obtain the full 3D camera rotation.

[3] The source code and experiment data are available at: http://cphoto.fit.vutbr.cz/semantic-orientation.

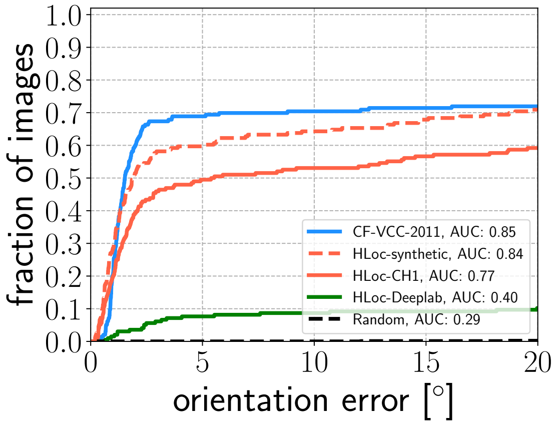


Figure 6.13: Results on CH1 dataset. Our **CF** framework using *automatic* segments + edges has superior accuracy compared to **HLoc** with original, *manually* refined horizon line from CH1 dataset (**HLoc-CH1**).

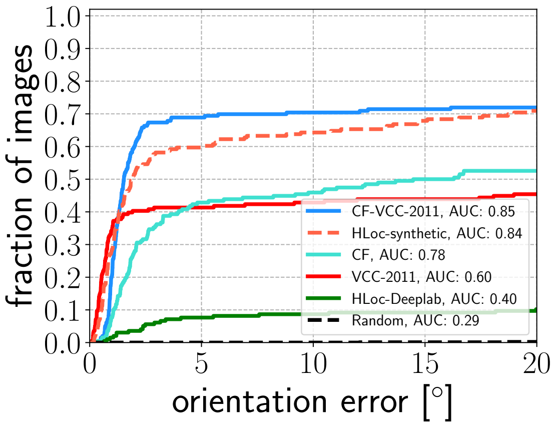


Figure 6.14: CH1 dataset: Comparison of the edge-based VCC-2011, our **CF** framework using semantic segments, and combination of both approaches. We use our **CF** framework to fuse semantic segments and edges together (**CF-VCC-2011**), which gives the best result.

We report the results on the GeoPose3K test set (Fig. 6.12), CH1 dataset [14] (Fig. 6.13), and Venturi Mountain dataset [142] (Tab. 6.2). First, we provide an upper bound of our *HLoc* implementation. We measure results with horizon lines rendered from the DEM with perspective projection (*HLoc-synthetic*). Second, we measure *HLoc* performance on queries with automatically segmented sky class using Deeplab (*HLoc-Deeplab*). Third, on the CH1 dataset, we use queries from the original publication [163], segmented with the help of the user (*HLoc-CH1*). According to our experiments, *HLoc* is quite sensitive to the quality of the segmentation – *HLoc-Deeplab* provides poor results compared to the *HLoc-synthetic* and *HLoc-CH1*. Our **CF** framework's performance is similar to *HLoc-synthetic* and is higher by a large margin compared to *HLoc-CH1*. Please note that our **CF** framework uses only **automatically** detected segments and edges. Compared to the *HLoc-Deeplab*, it scored significantly better on all three datasets. We conclude that the *HLoc* method depends on fine-grained horizon line segmentation, and it is not suitable for fully automatic processing. Our **CF** framework is much more robust to imprecisions in feature detection and achieves significantly better results compared to *HLoc* for automatically detected segment classes.

We further compare our method with the Robust silhouette map matching metric (m3D-2011) by Baboud *et al.* [15]. This non-linear metric penalizes crossings of query edges with synthetic depth discontinuities. The metric is accurate; however, it needs a reasonably small subset of candidate rotations since its computation time is enormous (hours

Resolution	Method	Avg. mean	Avg. stddev	F1	F2	F3	F4	F5	F6	J1	J2	J3	J4	J5	J6
low	**CF-VCC-2011-m3D** (ours)	**5.93**	**21.82**	1.82	**3.50**	30.26	**4.15**	**13.92**	4.02	**3.51**	1.20	1.31	**1.20**	**5.93**	**2.41**
	VCC-2011-m3D	21.06	44.20	**1.00**	6.01	**21.27**	116.87	41.30	**1.69**	132.92	**0.71**	**0.55**	**1.20**	41.04	4.46
	CF-VCC-2011 (ours)	34.19	41.75	6.67	5.00	100.11	132.06	51.42	39.95	7.41	6.75	23.87	8.85	55.39	19.01
high	**CF-VCC-2011-m3D** (ours)	**1.92**	**10.62**	2.57	3.68	**1.06**	1.57	**2.68**	**0.61**	4.54	1.26	**0.50**	**1.18**	**5.24**	**0.47**
	VCC-2011-m3D	2.88	14.72	1.49	8.94	1.27	6.25	4.42	1.18	5.17	1.08	**0.50**	**1.18**	6.29	0.66
	CF-VCC-2011 (ours)	12.42	32.44	**0.93**	**0.67**	85.68	**1.09**	21.18	2.45	**1.85**	**0.93**	8.32	1.42	41.65	0.75
-	HLoc-synthetic	28.0	50.54	52.73	1.84	11.54	36.08	4.17	10.21	115.54	86.08	6.01	4.11	3.84	40.85
-	HLoc-Deeplab	98.76	61.24	133.69	47.52	85.66	128.47	54.48	120.4	115.23	134.89	28.61	100.1	57.67	155.35
-	RFN_h – HOR [142]	1.23	1.24	-	-	-	-	-	-	-	-	-	-	-	-
-	SENSORS [142]	9.43	4.16	-	-	-	-	-	-	-	-	-	-	-	-

Table 6.2: Mean orientation error (in degrees) of the proposed method and its variants on Venturi Mountain dataset (video sequences F1 – F6, and J1 – J6). The last two rows refer to the reference results obtained with the help of device inertial sensors by Porzi *et al*. [142].

per query on the whole SO(3)). One can look at this metric as a geometric verification step; once the subset of candidate rotations is known, we can use this metric to verify and re-rank the best candidates. Tab. 6.2 illustrates that our **CF-VCC-2011** framework (segments + edges) already outperforms more complex Robust silhouette map matching metric (VCC-2011-m3D) on several Venturi sequences (F1, F2, F4, J1, and J2) on high resolution. On the other hand, our method's mean error is considerably higher than VCC-2011-m3D for the sequences F3, F5, and J5. These sequences are sparsely populated with synthetic semantic segment descriptions, which is attributed to the inaccuracy of the GIS database (OpenStreetMap). Additionally, in these sequences, the horizon line is often straight, which rapidly reduces its descriptivity.

However, when we use **CF-VCC-2011** as an initial estimate (search space reduction) for m3D-2011 [15], we improve the state-of-the-art result of m3D-2011, since it searches through the smaller number of outlier candidates. Considerable improvement has been achieved, especially at low resolution. The combination of m3D-2011 and our **CF** method (**CF-VCC-2011-m3D**) achieves a mean error of 5.93°, which is smaller by more than 70% compared to the original VCC-2011-m3D method (see Tab. 6.2, **CF-VCC-2011-m3D** vs. VCC-2011-m3D). Such an improvement is an important result since the proposed method is fast on low resolution (seconds per query). **CF-VCC-2011-m3D** achieved the most accurate result (1.92°) at high resolution. The improvement over the original method **VCC-2011-m3D** (2.88°) is 33%.

6.4 Chapter Summary

We proposed a novel method for camera orientation estimation in natural scenes based on semantic segmentation cues. To extract semantic segments from the query image, we utilized three state-of-the-art semantic segmentation methods and evaluated their suitability for the orientation estimation task. We used an extensive synthetic dataset, GeoPose3K, to train semantic segmentation methods to extract natural segments like forested areas, water bodies, sky segments, or glaciers.

Our experiments indicate that the semantic segments' boundaries are less informative than their areas, therefore complementing the information stored in edge maps. Using the proposed confidence-based fusion framework, we measured that semantic segments are more informative than edges. However, as the edges add complementary information to the estimation process, the *combination* of semantic segments and edges achieves the state-of-the-art result in camera orientation estimation on natural scenes.

Chapter 7

Matching Photographs with DEM Using Learned Cross-domain Descriptors

Augmented reality (AR) systems rely on approximate knowledge of physical geometry to facilitate the interaction of virtual objects with the physical scene, and tracking of the camera pose in order to render the virtual content correctly. In practice, a suitable scene approximation such as one or multiple planes, is tracked with the help of active depth sensors, stereo cameras, or multiview geometry from monocular video (*e.g.*, SLAM). All of these approaches are limited in their *operational range*, due to constraints related to light falloff for active illumination, and stereo baselines and camera parallax for multiview methods.

In this chapter, we propose a solution for outdoor *landscape-scale* augmented reality applications by registering the user's camera feed to large scale DEMs overlaid with a satellite orthophoto texture, see Fig. 7.1. As there is significant appearance variation between the DEM and the camera feed, we train a data-driven cross-domain feature descriptor that allows us to perform efficient and accurate feature matching. Using this approach, we can localize photos based on long-distance cues, allowing us to display large scale augmented reality overlays such as altitude contour lines, map features (roads and trails), or 3D created content, such as educational geographic-focused features. We can also augment long-distance scene content in images with DEM derived features, such as semantic segmentation labels, depth values, and normals.

Since modern mobile devices and many cameras come with built-in GPS, compass, and accelerometer, we could attempt to compute alignment from this data. Unfortunately, all of these sensors are subject to various sources of imprecision; *e.g.*, the compass suffers from magnetic variation (irregularities of the terrestrial magnetic field) as well as deviation (unpredictable irregularities caused by deposits of ferrous minerals, or even by random small metal objects around the sensor itself). This means that while the computed alignment is usually close enough for rough localization, the accumulated error over geographical distances results in visible mismatches in places such as the horizon line.

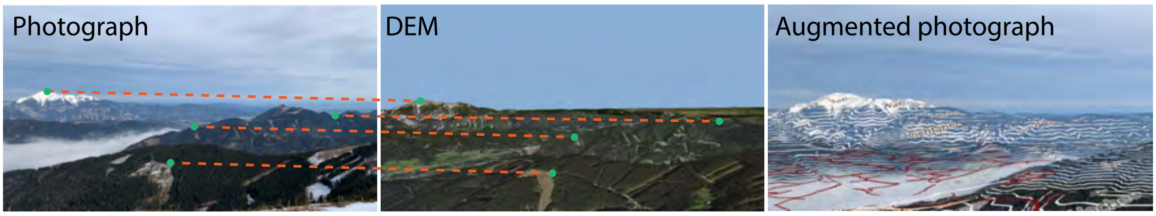


Figure 7.1: Our method matches a query photograph to a rendered digital elevation model (DEM). For clarity, we visualize only four matches (dashed orange). The matches produced by our system can then be used for localization, which is a key component for augmented reality applications. In the right image (zoomed-in for clarity), we render countour lines (white), gravel roads (red), and trails (black) using the estimated camera pose.

The key insight of our approach is that we can take advantage of a robust and readily available source of data, with near-global coverage, that is DEM models, in order to compute camera location using reliable, 3D feature matching based methods. However, registering photographs to DEMs is challenging, as both domains are substantially different. For example, even high-quality DEMs tend to have resolution too rough to capture local high-frequency features like mountain peaks, leading to horizon mismatches. Additionally, photographs have (often) unknown camera intrinsics such as focal length, exhibit seasonal and weather variations, foreground occluders like trees or people, and objects not present in the DEM itself, like buildings.

Our method works by learning a data-driven cross-domain feature embedding. We first use structure from motion (SfM) to reconstruct a robust 3D model from internet photographs, aligning it to a known terrain model. For reconstruction, we use our novel approach based on a direct matching of photographs with images rendered from the terrain model described in Section 5.3. We then render views at similar poses as photographs, which lets us extract cross-domain patches in correspondence, which we use as supervision for training. No 3D reconstruction is needed at test time, and features from the query image can be matched directly to renderings of the DEM.

Registration to DEMs only makes sense for images that observe a significant amount of content *farther* away than ca 100 meters. For this reason, we focus on mountainous regions, where distant terrain is often visible. While buildings would also provide a reasonable source for registration, we do not test on buildings, as building geometry is diverse, and 3D data and textures for urban areas are not freely available.

Our method is efficient and runs on a mobile device. As a demonstration, we developed a mobile application that performs large-scale visual localization to landscape features locally on a recent iPhone. We show that our approach can be used to refine localization when embedded device sensors are inaccurate.

Contributions. We propose a novel data-driven cross-domain embedding technique suitable for computing similarity between patches from photographs and a textured terrain model. To train our cross-domain descriptor based on a CNN, we propose a novel weakly supervised training scheme for positive/negative patch generation from the SfM reconstruction aligned with a DEM. We show that our novel embedding can be used for matching photographs to the terrain model to estimate respective camera position and orientation.[1] We also implement our system on the iPhone, showing that mobile large scale localization is possible on-device.

7.1 Related Work

7.1.1 Local descriptors

Most classical solutions to correspondence search involve using descriptors computed from local windows around feature points. These descriptors can be either hand-designed, *e.g.*, SIFT [121], SURF [20], ORB [155], or learned end-to-end [221, 126, 192, 60, 48]. The key difference between our method and these is that we train our method in a cross-domain scenario, in which we match two images with different appearance, *e.g.*, the photograph and the outdoor image rendered using digital elevation model. While our method is also a local descriptor, it is designed to deal with additional appearance and geometry differences, which is not the case for these methods.

Of these, HardNet++ [126] and D2Net [48] have been trained on outdoor images (HardNet on Brown dataset and HPatches, D2Net on Megadepth [115] which contains 3D reconstructed models in the European Alps and Yosemite). Since it is possible that a powerful enough single-domain method might be able to bridge the domain gap (as demonstrated for D2Net and sketches), and these two methods are compatible with our use-case, we chose them as baselines to compare with our method.

7.1.2 Cross-domain matching

A large body of research work has been devoted to alignment of multi-sensor images [205, 82, 92] and to modality-invariant descriptors [176, 35, 74, 174, 105]. These efforts often focus on optical image alignment with *e.g.*, its infra-red counterpart. However, our scenario is much more challenging, because we are matching an image with a *rendered* DEM where the change in appearance is considerable.

With the advent of deep-learning, several CNN-based works on matching multimodal patches emerged and outperformed previous multimodal descriptors [2, 3, 49, 61, 19].

[1] Code & data are available at: `http://cphoto.fit.vutbr.cz/LandscapeAR`

However, cross-spectral approaches [2, 3, 49, 19] need to account only for rapid visual appearance change, compared to our scenario, which needs to cover also the differences in scene geometry, caused by limited DEM resolution. On the other hand, RGB to depth matching approaches, such as Georgakis et al. [61] lack the texture information and need to focus only on geometry, which is not our case.

7.2 Camera Pose Estimation Using Cross-domain Descriptor

Our goal is to estimate the camera pose of a single query image with respect to the synthetic globe, which can be cast as a standard Perspective-*n*-Point problem [55] given accurate correspondences. The main challenge is to establish correspondences between keypoints in the query photograph and a rendered synthetic frame, two domains very different appearance-wise. We bridge this appearance gap by training an embedding function that projects local neighborhoods of keypoints from either domain into a unified descriptor space. Two cross-domain keypoints are assumed to correspond if the L2 distance of their descriptors is mutually closest.

7.2.1 Weakly Supervised Cross-domain Patch Sampling

While the rendered image is assumed to contain a similar view as the photograph, it is not exact. Therefore, our embedding function should be robust to slight geometric deformations caused by viewpoint change, weather and seasonal changes, and different illumination. Note that these phenomena do not occur only in the photograph, but also in the orthophoto textures. Previous work on wide baseline stereo matching, patch verification, and instance retrieval illustrate that these properties could be learned directly from data [7, 126, 153, 48]. For efficient training process, an automatic selection of corresponding (positive) and negative examples is crucial. In contrast with other methods, which rely on the reconstructed 3D points [126, 48] dependent on a keypoint detector, we instead propose a weakly supervised patch sampling method completely independent of a preexisting keypoint detector to avoid any bias that might incur. Being independent on a keypoint detector is an important and desirable property for our cross-domain approach, since (I) the accuracy of existing keypoint detectors in the cross-domain matching task is unknown, (II) our embedding function may be used with any keypoint detector in the future without the need for re-training.

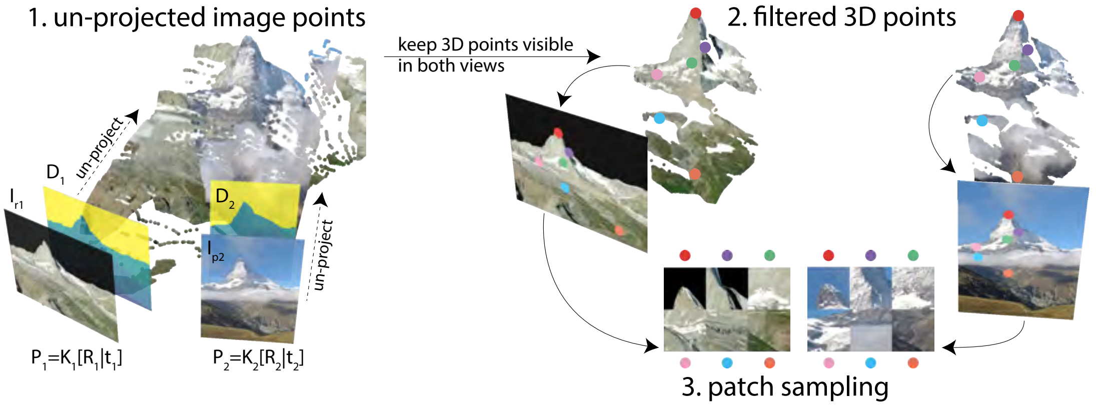


Figure 7.2: A method for sampling corresponding pairs of cross-domain patches. 1. For a pair of images I_{r1} (render), I_{p2} (photograph), 2D image points are un-projected into 3D using the rendered depth maps D_1, D_2, and the ground truth camera poses P_1, P_2, respectively. 2. Only points visible from both views are kept. 3. A randomly selected subset of 3D points is used to form patch centers, and corresponding patches are extracted. Image credit: John Bohlmeyer (`https://flic.kr/p/gm3xwP`).

Each photograph in our dataset contains ground truth camera pose $P = \mathbf{K}[\mathbf{R}|t]$ transforming the synthetic world coordinates into the camera space. For each photograph I_{p1}, we render a synthetic image I_{r1} and a depth map D_1, see Fig. 7.2. We pick all pairs of cameras with at least 30 corresponding 3D points in the SfM reconstruction. For each pair, the camera pose and depth map are used to un-project all image pixels into a dense 3D model (Fig. 7.2-1.). Next, for each domain, we keep only the 3D points visible in both views (Fig. 7.2-2.). Finally, we uniformly sample N random correspondences (Fig. 7.2-3.), each defining the center of a local image patch.

7.2.2 Architecture

To account for the appearance gap between our domains, we employ a branched network with one branch for each of the input domains followed by a shared trunk. A description of the architecture is shown in Fig. 7.3. The proposed architecture is fully convolutional and has a receptive field of 63 px. To get a single descriptor, we use an input patch of size 64 × 64 px. We use neither pooling nor batch normalization layers. Similarly to HardNet [126], we normalize each input patch by subtracting its mean and dividing by its standard deviation. Thanks to the structure of our task formulation and the simplicity of the chosen architecture, our network is quite compact and contains only 358,976 trainable parameters, compared to D2Net [48], which contains more than 7.6 million of trainable parameters. The small size allows our architecture to be easily deployed to a mobile device like the iPhone, enabling a broader scale of applications.

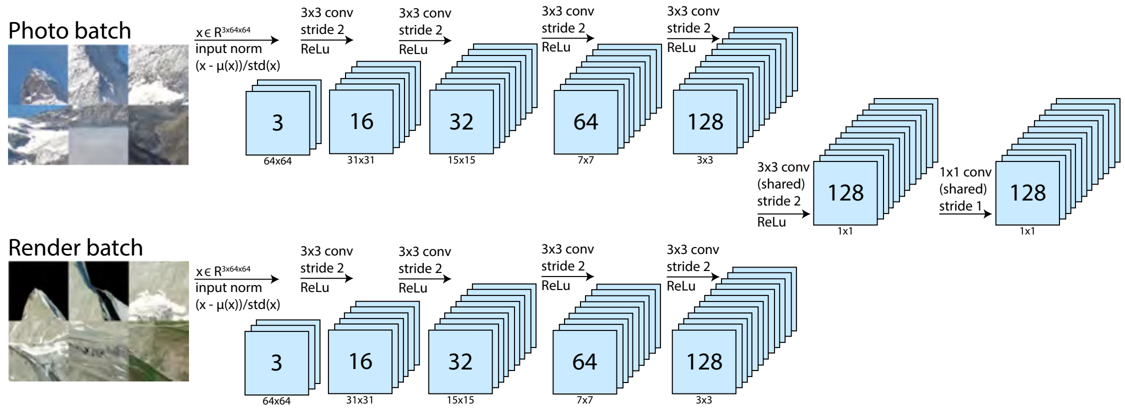


Figure 7.3: Architecture of our two branch network with partially shared weights for cross-domain descriptor extraction. Photo and render branches contain four 3x3 2D convolutions with stride 2; weights are not shared between branches. The last two convolutions form a trunk of the network with shared weights to embed both domains into a single space. Output is 128-d descriptor. Either one or the other branch is used, each branch is specific for its own domain. Image credit: John Bohlmeyer (`https://flic.kr/p/gm3xwP`).

7.2.3 Training

We use a standard triplet loss function adjusted to our cross-domain scenario:

$$L(\mathfrak{a}^h, \mathfrak{p}^r, \mathfrak{n}^r) = \sum_i \max(\|\mathfrak{a}_i^h - \mathfrak{p}_i^r\|_2 - \|\mathfrak{a}_i^h - \mathfrak{n}_i^r\|_2 + \alpha, 0)), \tag{7.1}$$

where $\mathfrak{a}$, $\mathfrak{p}$, $\mathfrak{n}$ denotes a mini-batch of anchor, positive, and negative descriptors, respectively. The superscript denotes the modality from which the descriptor was calculated: photograph (h) calculated using the photograph branch of the network, or render (r) calculated using the render branch of the network. The α denotes the margin.

Previous work on descriptor learning using the triplet loss function [126] illustrated the importance of sampling strategy for selecting negative examples. In this solution, for each patch in a mini-batch, we know its 3D coordinate in an euclidean world space $x(p_j) \in \mathbb{R}^3$. Given a mini-batch of anchor and positive descriptors $\mathfrak{a}_i^h, \mathfrak{p}_i^r, i \in [0, N]$ where N is a batch size, we first select subset of *possible* negatives $\overline{\mathfrak{n}^r}$ from all positive samples within a current batch, which are farther than m meters from the anchor:

$$\overline{\mathfrak{n}^r} = \left\{ \mathfrak{p}_j^r \,\middle|\, \left[\|x(\mathfrak{p}_j^r) - x(\mathfrak{a}_i^h)\|_2\right] > m \right\}. \tag{7.2}$$

In HardNet [126], for each positive only a hardest negative from the subset of possible negatives should be selected. However, we found that this strategy led the embedding function to collapse into a singular point. Therefore, we propose an adaptive variant of hard negative sampling inspired by a prior off-line mining strategy [73], modified to operate on-line.

We introduce a curriculum to increase the difficulty of the randomly sampled negatives during training. In classic hard negative mining, for each anchor descriptor $\mathfrak{a}_i$ we randomly choose a possible negative descriptor $\overline{\mathfrak{n}_j}$ as a negative example $\mathfrak{n}_j$, if and only if the triplet loss criterion is violated:

$$\|\mathfrak{a}_i - \overline{\mathfrak{n}_j}\|_2 < \|\mathfrak{a}_i - \mathfrak{p}_i\|_2 + \alpha, \tag{7.3}$$

where we denote $\mathfrak{a}_i = \mathfrak{a}_i^h$ as an anchor descriptor calculated from a photo patch using the photo encoder, and similarly for $\overline{\mathfrak{n}_j} = \overline{\mathfrak{n}_j^r}$, and $\mathfrak{p}_i = \mathfrak{p}_i^r$ encoded by the render encoder. We build on this, and for each anchor descriptor $\mathfrak{a}_i$, randomly choose a possible negative descriptor $\overline{\mathfrak{n}_j}$ as a negative example $\mathfrak{n}_j$ iff:

$$\|\mathfrak{a}_i - \overline{\mathfrak{n}_j}\|_2 < d^+ - (d^+ - (\mathfrak{n}_{\min} + \epsilon)) \cdot \lambda, \tag{7.4}$$

where λ is a parameter in $[0, 1]$ defining the difficulty of the negative mining, $\epsilon \to 0^+$ is a small positive constant, d^+ is the distance between anchor and positive plus margin:

$$d^+ = \|\mathfrak{a}_i - \mathfrak{p}_i\|_2 + \alpha, \tag{7.5}$$

and $\mathfrak{n}_{\min}$ is the distance between the anchor and the hardest negative:

$$\mathfrak{n}_{\min} = \min_j \|\mathfrak{a}_i - \overline{\mathfrak{n}_j}\|_2. \tag{7.6}$$

Intuitively, when $\lambda = 0$, Eq. 7.4 is reduced to random hard negative sampling defined in Eq. 7.3, and when $\lambda = 1$, the Eq. 7.4 is forced to select $\overline{\mathfrak{n}_j}$ as a negative only if it is equal to the hardest negative $\mathfrak{n}_{\min}$, reducing the sampling method to HardNet [126].

The parameter λ allows us to select harder negatives throughout the training. We start training with the $\lambda = 0$ and increase λ by 0.05 with each 10,000 steps up to a maximum hardness. Once maximum hardness is reached, we keep it constant until the end of the training. We experimentally found that a maximum of $\lambda = 0.23$ worked well for our data, with the margin set to $\alpha = 0.2$, and minimum distance in 3D was set to $m = 50\,\text{m}$. We used minibatch size of 300 patches, learning rate 10^{-5}, and ADAM optimizer. To prevent overfitting we used early stopping using validation set; the network was trained for 21 epochs using 1.2 million training steps.

So far, we defined our loss function to be a cross-domain triplet loss, having an anchor as a *photograph*, and the positive and negative patches as *renders*. However, this loss function optimizes only the distance between the *photograph* and *render* descriptors. As a result, we use a variant with auxiliary loss functions optimizing also the distances between *photo-photo* and *render-render* descriptors:

$$L_{\text{aux}} = L(\mathfrak{a}^h, \mathfrak{p}^r, \mathfrak{n}^r) + L(\mathfrak{a}^h, \mathfrak{p}^h, \mathfrak{n}^h) + L(\mathfrak{a}^r, \mathfrak{p}^r, \mathfrak{n}^r). \tag{7.7}$$

As we illustrate by our experiments, this variant performs the best in the cross-domain matching scenario.

7.2.4 Pose Estimation

We illustrate the performance of our descriptor on a camera pose estimation task from a single query image. For each query image, we render a fan of 12 images with field-of-view FOV=60° rotated by 30° around the vertical axis, similarly to Fig. 5.3-2. The input photograph is scaled by a scale factor s proportional to its FOV θ: $s = (\theta \cdot M)/(\pi \cdot I_w)$, where M is the maximum resolution corresponding to FOV=180° and I_w is the width of the image. We use the SIFT keypoint detector (although any detector could be used), take a 64×64 px patch around each keypoint, and calculate a descriptor using our method.

We start by finding top-3 candidates between images of the rendered fan. For this purpose, we use a simple voting strategy: for each rendered image we calculate the number of mutual nearest neighbor matches with the input photograph. We use top-3 candidates, since the photograph is unlikely to span over more than three consecutive renders, covering a FOV of 120°. For each top candidate, we un-project the 2D points from the rendered image to 3D using rendered camera parameters and a depth map; then we compute the full camera pose of the photograph with respect to the 3D coordinates using OpenCV implementation of the EP*n*P [110] algorithm with RANSAC. From the three output camera poses, we select the *best pose*, which minimizes the reprojection error while having a reasonable number of inliers; if any candidate poses have more than $N = 60$ inliers, we select the one with the lowest reprojection error. If none are found, we lower the threshold N and check for the *best pose* in a new iteration. If there is no candidate pose with at least $N = 20$ inliers, we end the algorithm as unsuccessful.

Finally, we reproject all the matched 3D points—not only inliers—into the camera plane using the *best pose*, and select those within the frame. We match the 2D image keypoints with the selected 3D points and calculate the *refined pose* using EP*n*P inside a RANSAC loop.

7.3 Experiments

In this section, we introduce the experiments based on our pose estimation pipeline utilizing our cross-domain descriptor to compute the camera pose relative to the terrain model. We present majority of the results as cumulative error plots, where we count the fraction of images localized below the distance or rotation error threshold; for the detailed description of the evaluation protocol, see Sec. 3.2.4. An ideal system is located at the top-left corner, where all the images are localized with zero distance and rotation errors. Throughout the experiments section, we denote our architecture and its variants trained on our training dataset as **Ours-***. In addition, we report results for a larger single-branch architecture based on VGG-16 fine-tuned on our data (denoted as **VGG-16-D2-FT**). Similarly to D2Net, we cut the VGG-16 at conv 4-3, load the D2Net weights, and add two more convolutional layers to subsample the result descriptor to 128 dimensions. The newly added layers as well as the conv 4-3 were fine-tuned using our training method and data.

We compare our methods with state-of-the-art deep local descriptors or matchers: HardNet++ [126], D2Net [48], and NCNet [153], which we use with original weights. Initially, we tried to train the HardNet and D2Net methods on our training dataset using their original training algorithms, but the results did not exhibit any improvements. We did not try to train the NCNet, since this method outputs directly matches and consumes many computational resources, which is undesirable with our target applications capable of running on a mobile device.

7.3.1 Test Datasets

To evaluate our method in a cross-domain scenario, we use the publicly available dataset GeoPose3K [A1] spanning an area of the European Alps. We used the standard publicly available test split of 516 images [A3]. We note that we were very careful while constructing our training dataset *not* to overlap with the test area of the GeoPose3K dataset. To illustrate that our method generalizes over the borders of the European Alps, on which it was trained, we also introduce three more test sets: *Nepal* (244 images), *Andes Huascaran* (126 images), and *Yosemite* (644 images). We constructed the *Nepal* and *Yosemite* datasets using SfM reconstruction using SIFT keypoints aligned to the terrain model with the iterative closest points algorithm described in Sec 5.2. The *Huascaran* dataset has been constructed using the direct cross-domain reconstruction algorithm described in Sec. 5.3. Please note that this particular dataset may therefore be biased towards D2Net [48] matchable points, while *Nepal* and *Yosemite* datasets might be biased towards SIFT matchable points. Unlike the training images, camera poses in the test sets were manually inspected, and outliers were removed.

7.3.2 Ablation Studies

Best Pose and Refined Pose. We study the behavior of our cross-domain pose estimation approach on the GeoPose3K dataset, on which we evaluate the *best pose* (solid) and the *refined pose* (dashed) for three different embedding algorithms, as illustrated in Fig. 7.4. In the left plot, we can see that the *refined pose* improves over the *best pose* for both HardNet++ and our method for well-registered images (up to distance error around 300 m), whereas it decreases result quality with D2Net. We hypothesize that this is because in the pose refinement step, the descriptor needs to disambiguate between more distractors compared to the case of the best pose, where a single photograph is matched with a single rendered

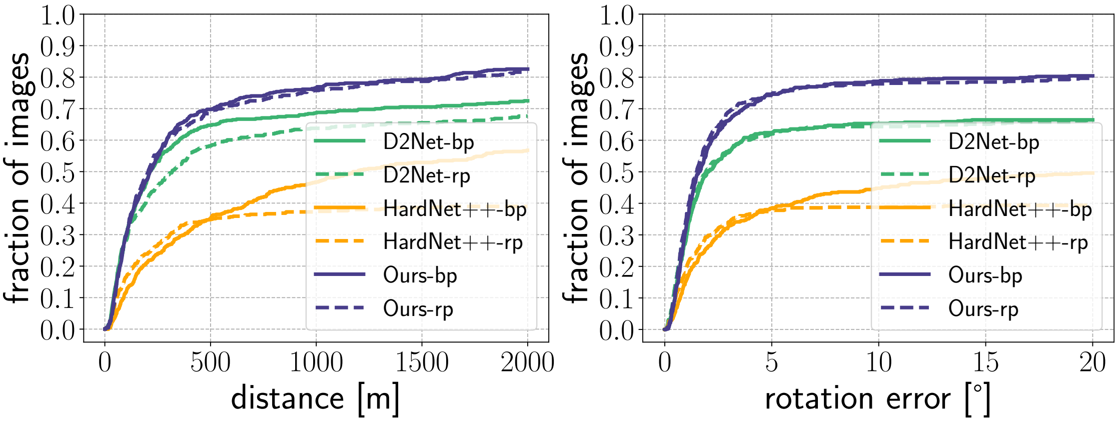


Figure 7.4: Comparison of the *best pose* (bp) and the *refined pose* (rp) using different descriptors on GeoPose3K using *cross-domain* matches between the query photograph and synthetically rendered panorama. **Left:** translation error, **right:** rotation error.

	Position error [m]					**Rotation error [°]**				
Method	100	300	500	700	900	1	3	5	7	9
	Cumulative fraction of photographs									
Ours-RSH	0.29	0.53	0.61	0.65	0.67	0.34	0.56	0.60	0.63	0.64
Ours-ASH	**0.30**	**0.54**	**0.63**	**0.67**	**0.70**	**0.39**	**0.60**	**0.65**	**0.68**	**0.69**

Table 7.1: Comparison of two training strategies of our *cross-domain* network on the pose estimation task on GeoPose3K dataset using *cross-domain* matches between the query photograph and the rendered panorama. The higher number the better. Adaptive semihard (ASH) performs better than random semihard (RSH).

image, and D2Net seems to be more sensitive to these distractors than other approaches. Furthermore, the right plot in Fig. 7.4 shows that the rotation error is improved on the refined pose for all three methods up to the threshold of 5°. Since points from multiple rendered views are already matched, the subsequent matching step covers a wider FOV, and thus a more reliable rotation can be found. For the following experiments, we use the *refined pose*, which seems to estimate camera poses with slightly better accuracy in the low-error regime.

Random Semi-hard and Adaptive Semi-hard Negative Mining. We analyze the difference between the baseline random semi-hard negative mining and adaptive semi-hard negative mining in Tab. 7.1. The experiment illustrates that adaptive semi-hard negative mining improves the random semi-hard negative mining baseline in both position and orientation errors, so we use it in all experiments.

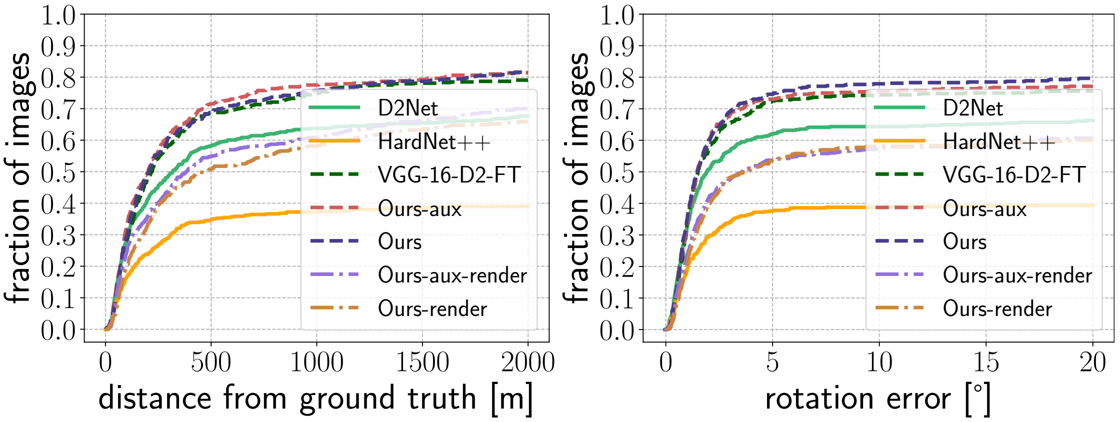


Figure 7.5: Comparison of variants of our network with HardNet++ and D2Net for pose estimation task on GeoPose3K using *cross-domain* matches between query photograph and synthetically rendered panorama. **Left:** translation error, **right:** rotation error.

Auxiliary Loss. Our network trained with the auxiliary loss function performs the best in the cross-domain scenario evaluated on the GeoPose3K dataset (Fig. 7.5, see Ours-aux). On this task, it outperforms the cross-domain variant of our network trained with the basic loss function (Ours). We also report our network's result using a single encoder for both domains (Ours-render) which is consistently worse than the cross-domain variant. Furthermore, we see here that our network significantly outperforms both D2Net and HardNet++ in this task.

Stability With Respect to DEM Sampling Density. One question is how close does our DEM render has to be to the true photo location for us to still find a correct pose estimate. To evaluate this, for each query photograph (with known ground truth location), we render a synthetic reference panorama offset from the photo location by a random amount (the "baseline"), sampled from a Gaussian distribution with parameters $\mathcal{N}(0\,\text{m}, 1000\,\text{m})$. We then estimate the pose of the query photograph by registering it with the render and compare the predicted location to the known ground truth location. In Fig. 7.6-left, we show the percentage of cases where the distance from ground truth to the predicted location was predicted to be less than the baseline. This gives us a measure, for example, of how incorrect the GPS signal from a photo could be such that our approach improves localization. With low baselines, we see that the geometry mismatch to the DEM dominates, and the position is difficult to improve on. With baselines over 200 m, we are able to register the photo, and then performance slowly degrades with increased baselines as matching becomes difficult. Fig. 7.6-right shows that the cross-over point where the position no longer improves over reference is around 700 m.

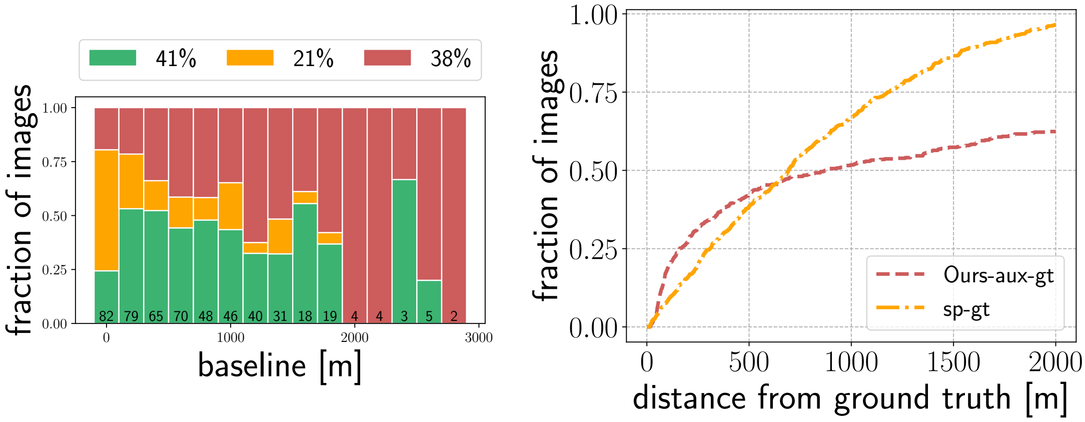


Figure 7.6: Evaluation of robustness to the baseline. **Left:** Fraction of improved (green), worsen (yellow), and failed (red) positions when matching query photo to a synthetic panorama as a function of the baseline. The baseline is the distance between the ground truth position and a *reference position* generated by adding a Gaussian noise $\mathcal{N}(0\,\text{m}, 1000\,\text{m})$ to the ground truth position. We consider the position improved when the estimated distance to ground truth is less than the baseline. The numbers at the bottom of each bar give the total number of images within each bar. **Right:** Cumulative fraction of query photos with an estimated position less than a given distance from ground truth (Ours-aux in pink) versus the cumulative fraction of *reference positions* within a given distance of ground truth (sp-gt in yellow). The pink line above the yellow line means our method improves over the sampled *reference position* at that baseline.

7.3.3 Comparison with State-of-the-Art

We compare our two-branch method and single-branch method based on VGG-16 with three state-of-the-art descriptors and matchers: HardNet [126], D2Net [48], and NCNet [153] in four different locations across the Earth. According to the results in Fig. 7.7, our two-branch method trained with auxiliary loss function (Ours-aux) exhibits the best performance on *GeoPose3K*, *Nepal*, and *Yosemite* datasets. The only dataset where our two-branch architecture is on-par with D2Net is *Andes Huascaran* (where the ground truth was created by D2Net matching) and where the single-branch VGG-16 architecture trained using our method and data performs the best. This result is probably due to differences in the orthophoto texture used to render synthetic images. The larger, pre-trained VGG-16 backbone has most likely learned more general filters than our two-branch network, which was trained solely on our dataset.

Additionally, we add a comparison with respect to the number of inliers given its month in the year, shown in Fig. 7.8. We may observe that more photographs across all datasets are usually captured during the summer and early autumn months (June–

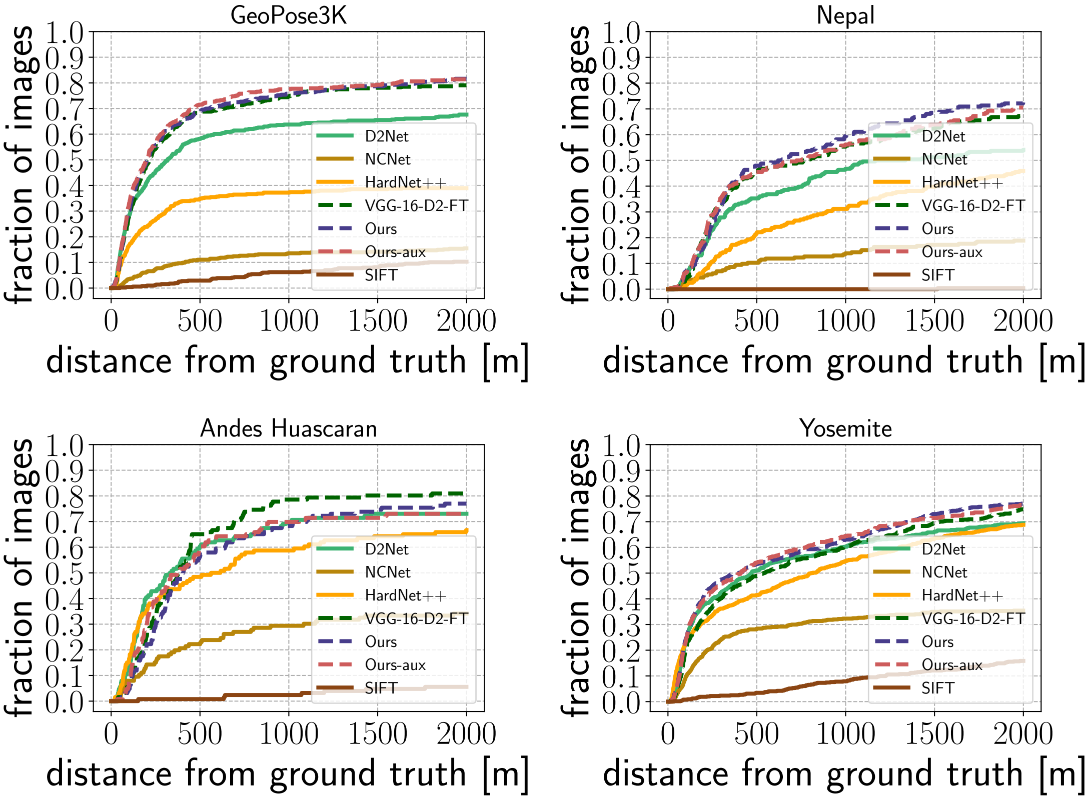


Figure 7.7: Comparison of our method with state-of-the-art descriptors in four different locations across the Earth. Our method (dashed red and blue) outperforms HardNet [126] on all datasets and D2Net [48] on GeoPose3K, Nepal and Yosemite. Our method seems to be on par with D2Net on Andes Huascaran dataset which has significantly less precise textures (from ESA RapidEye satellite) in comparison to other datasets.

October), see the dashed blue line. This observation correlates with the counts of inliers of all methods in the comparison—higher amounts of inliers are more likely in the summer photographs. On GeoPose3K and Nepal, our method trained with auxiliary loss functions (see red line) typically produces more inliers than D2Net and HardNet++. On Andes Huascaran and Yosemite, D2Net and HardNet++ are generally able to find more inliers than our method. This fact illustrates that images from the Nepal dataset are likely to have features similar to the Alps training set, which is less the case for Yosemite and Andes Huascaran. Moreover, Andes Huascaran is rendered using a different orthophoto texture (RapidEye satellite) and was created using D2Net matches, giving an advantage to this method.

Mean and median counts of inliers for each method and dataset are illustrated in Tab. 7.2. Similarly to the per-month number of inliers, we see that our method retrieves

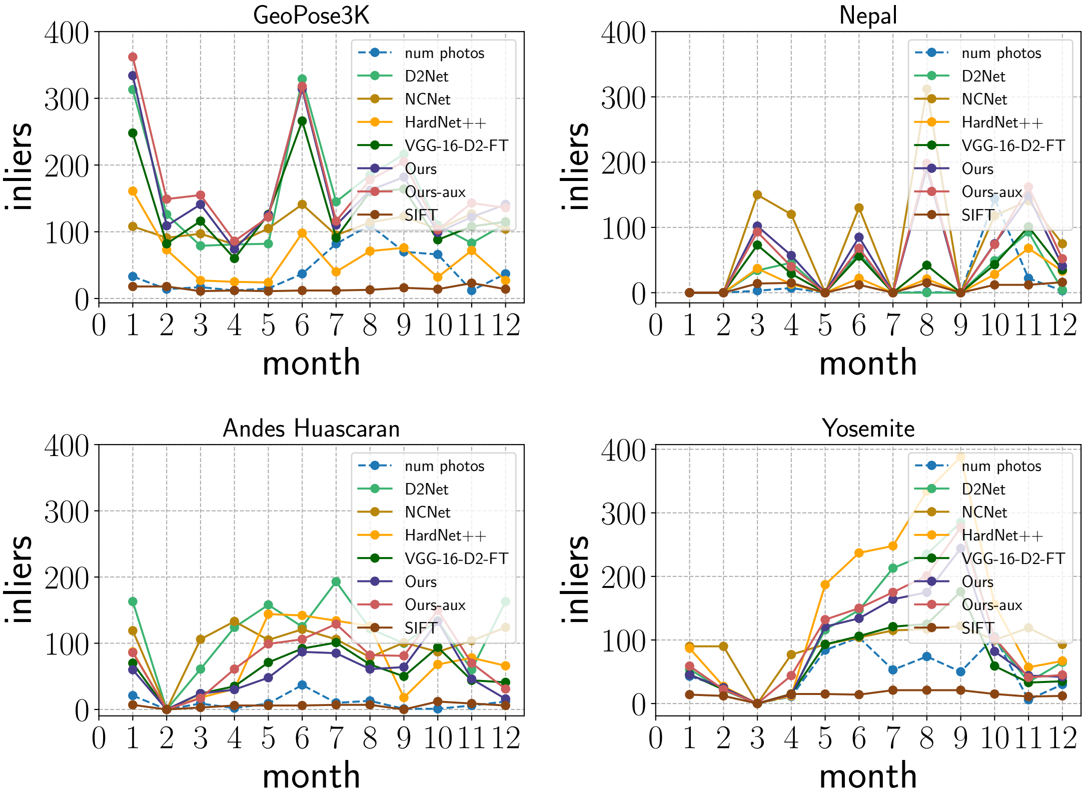


Figure 7.8: Comparison of our method with state-of-the-art with respect to number of inliers given the month in four different locations across the Earth. Higher is better.

	GeoPose3K		Nepal		Andes Huascaran		Yosemite	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
D2Net	175.72	69.50	55.41	33.00	**134.01**	**101.00**	142.84	21.00
~~NCNet~~	~~110.81~~	~~94.50~~	~~124.71~~	~~103.00~~	~~113.17~~	~~94.00~~	~~103.13~~	~~83.00~~
HardNet++	64.27	0.00	29.97	16.50	107.56	54.00	**207.66**	18.00
VGG-16-D2-FT	144.73	69.50	51.95	22.00	71.89	63.50	91.66	19.00
Ours	166.53	83.00	**84.52**	**49.50**	63.33	48.00	121.34	22.50
Ours-aux	**178.85**	**86.50**	80.14	43.50	86.79	80.00	137.70	**24.00**

Table 7.2: Comparison of our method with state-of-the-art with respect to number of inliers in four different locations across the Earth. The larger number the better, best performing algorithms are in bold. Although NCNet is able to get many inliers compared to other algorithms, we measured low amount of correctly localized images (see Fig 7.7–NCNet), and therefore we removed it from this comparison.

Method	Verification						Matching			Retrieval 20k		
	∗	◆	∗	◆	∗	◆						
	Easy		Hard		Tough		Easy	Hard	Tough	Easy	Hard	Tough
HardNet++	**0.986**	**0.979**	**0.974**	**0.962**	**0.939**	**0.919**	**0.730**	**0.582**	**0.401**	**0.792**	**0.677**	**0.492**
D2Net	0.810	0.788	0.721	0.700	0.666	0.646	0.387	0.172	0.075	0.545	0.312	0.179
VGG-16-D2-FT	0.866	0.834	0.770	0.734	0.706	0.671	0.168	0.459	0.017	0.292	0.122	0.062
Ours-photo	0.906	0.877	0.812	0.776	0.738	0.701	0.278	0.094	0.037	0.421	0.193	0.101
Ours-render	0.899	0.868	0.811	0.774	0.740	0.703	0.222	0.070	0.026	0.375	0.173	0.089
Ours-aux-photo	0.925	0.902	0.828	0.796	0.747	0.713	0.382	0.152	0.065	0.508	0.255	0.135
Ours-aux-render	0.956	0.942	0.915	0.892	0.857	0.830	0.453	0.231	0.112	0.556	0.326	0.181

Table 7.3: Comparison of variants of our network to HardNet++ and D2Net on the full HPatches dataset [17] (single domain). For D2Net, we used the dense feature extractor which results in 15x15 descriptors per 65px^2 patch, from which only the central descriptor was used. Higher is better in all tasks. HardNet++ perform the best, from our methods the render branch trained with auxiliary loss gives second best result (see bottom line). ∗ DiffSeq; ◆ SameSeq [17].

the most inliers on GeoPose3K and Nepal datasets, while D2Net and HardNet++ retrieve more inliers on Andes Huascaran and Yosemite, respectively. However, it seems that the inlier increase of HardNet++ on the Yosemite dataset is caused by few images with a large number of inliers, since the mean of HardNet++ is the largest, but the median is not—in fact, our method was able to get the largest median in the number of inliers on this dataset.

7.3.4 Auxiliary Loss Functions in Single-domain Scenario

According to our experiment, the auxiliary loss function defined in Eq. 7.7 brings further improvement over the basic variant of the cross-domain triplet loss defined in Eq. 7.1. To illustrate this, we evaluated each branch of our network on the single domain HPatches dataset [17] and compared it with HardNet++ and D2Net in Tab. 7.3 on three tasks—patch verification, matching, and instance retrieval. The symbols ∗ and ◆ denote DiffSeq (negative pairs are formed by patches from different sequences) and SameSeq (negative pairs are formed by patches from the same sequence), respectively—for its exact definition, please see the HPatches paper [17]. Please note that the HPatches benchmark is a single domain dataset containing only photographs, which is not compatible with the design of our architecture; moreover, our architecture was trained for a much more specific task than the competitors. Therefore, we needed to evaluate our network twice—once for each branch. HardNet++ exhibits superior performance over other methods on HPatches (see the first line in bold in Tab. 7.3), while on our cross-domain scenario it exhibits worse performance than our method (see Fig. 7.7). This illustrates that our cross-domain scenario is different from the single-domain one. On HPatches, the variant of our network trained with auxiliary loss function outperforms the variant trained with basic triplet loss, which is consistent with the comparison on our cross-domain datasets. Interestingly, the best performing variant of our method is the render branch trained with auxiliary loss functions (see the last line of Tab. 7.3 in bold). This result is most probably caused by the fact that in our train dataset, the rendered images are always aligned perfectly, unlike the photographs, which eventually can contain outliers.

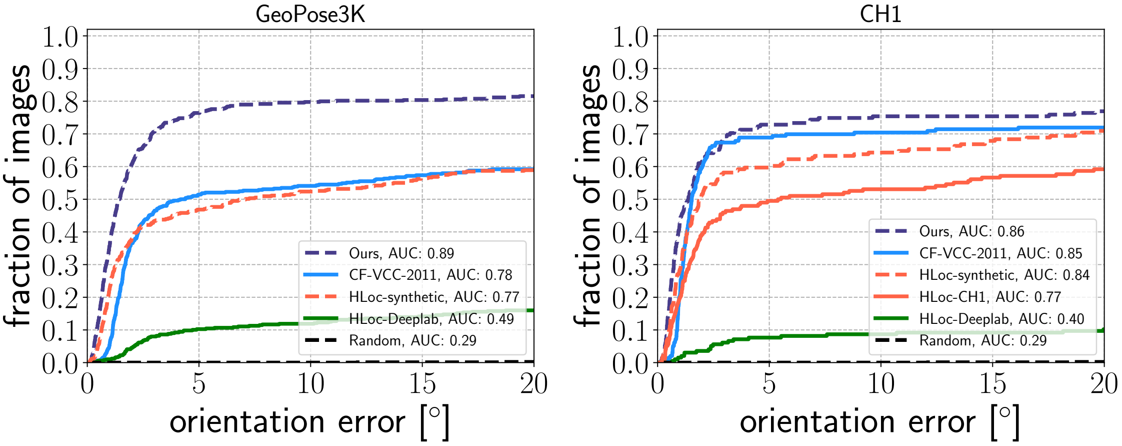


Figure 7.9: Comparison on the task of camera orientation estimation using our keypoint-based method (Ours, dashed dark blue) with horizon-line localization (HLoc, our reimplementation of [163], in orange and green) and our method based on semantic segments and edges (CF-VCC-2011, light blue, presented in Chapter 6) on GeoPose3K (left), and CH1 (right) dataset.

7.3.5 Comparison with Camera Orientation Estimation Methods

In Chapter 6, we proposed a novel method for 3-DOF camera orientation estimation based on a combination of edges and semantic segmentation called Confidence Fusion (CF). We compared it with an approach based on a horizon line, HLoc [163]. Let us now compare those 3-DOF camera orientation approaches with our novel 6-DOF, keypoint-based camera pose estimation method. Please note that the CF and HLoc methods which estimate camera orientation use the ground truth position as an input. Our keypoint-based camera pose estimation method solves a more complex task since it estimates both camera position and rotation. Therefore it is at a slight disadvantage in this comparison. On the other hand, the CF and HLoc methods use the rendered DEM *without* the satellite texture, which is needed for our keypoint-based camera pose estimation method.

In Fig. 7.9, we compare our keypoint-based camera pose estimation method with a Confidence Fusion method combining semantic segmentation and edge features (CF-VCC-2011) using a spherical cross-correlation described in Chap. 6, denoted in light blue. We

Method	Avg. mean	Avg. stddev	F1	F2	F3	F4	F5	F6	J1	J2	J3	J4	J5	J6
Ours	19.2	41.86	**0.62**	40.92	104.53	**1.03**	23.01	3.79	**0.85**	**0.84**	19.21	1.41	54.62	0.8
CF-VCC-2011-m3D	**1.92**	**10.62**	2.57	3.68	**1.06**	1.57	**2.68**	**0.61**	4.54	1.26	**0.50**	**1.18**	**5.24**	**0.47**
CF-VCC-2011	12.42	32.44	0.93	**0.67**	85.68	1.09	21.18	2.45	1.85	0.93	8.32	1.42	41.65	0.75
VCC-2011-m3D [15]	2.88	14.72	1.49	8.94	1.27	6.25	4.42	1.18	5.17	1.08	**0.50**	**1.18**	6.29	0.66
HLoc-synthetic	28.0	50.54	52.73	1.84	11.54	36.08	4.17	10.21	115.54	86.08	6.01	4.11	3.84	40.85
HLoc-Deeplab	98.76	61.24	133.69	47.52	85.66	128.47	54.48	120.4	115.23	134.89	28.61	100.1	57.67	155.35
RFN_h – HOR [142]	1.23	1.24	-	-	-	-	-	-	-	-	-	-	-	-
SENSORS [142]	9.43	4.16	-	-	-	-	-	-	-	-	-	-	-	-

Table 7.4: Comparison of our keypoint-based method with camera orientation estimation methods based on edges and semantic-segmentation on Venturi Mountain dataset [142] (video sequences F1–F6, and J1–J6). The last two rows refer to the reference results obtained with the help of device inertial sensors by Porzi *et al.* [142].

may also see the performance of a horizon-based localization method HLoc [163], which we implemented (see Chap. 4, 6), denoted in orange and green colors. HLoc-synthetic (dashed orange) uses a synthetically rendered horizon line (effectively illustrating the upper bound of the method); HLoc-CH1 (solid orange) uses horizon line segmentation from the CH1 dataset (refined manually [163]), and HLoc-Deeplab uses the horizon line segmented automatically using DeepLab semantic segmentation trained on GeoPose3K training set. Please note that we cannot use GeoPose3K or CH1 datasets for a fair comparison with the edge-based non-linear metric originally proposed by Baboud *et al.* [15] since the camera orientation ground truth has been set with the help of this method. To compare our keypoint-based approach with this non-linear edge-based metric, we use the Venturi Mountain dataset [142].

Our keypoint-based method is a clear winner on the GeoPose3K dataset (Fig. 7.9-left) and is on-par with the Confidence Fusion method on the CH1 dataset (Fig. 7.9-right). We hypothesize that the main reason is that GeoPose3K is a more diverse test which contains images in which distinct keypoints can be detected. CH1 dataset, on the other hand, has been composed as a benchmark for a horizon-line based method, and it often contains photographs with a distinct horizon line, but no distinct keypoints. This is illustrated by the fact that methods which use horizon line as a feature, Hloc-synthetic (dashed orange), and CF-VCC-2011 (solid blue) perform better on the CH1 dataset than on the GeoPose3K dataset. Although CH1 dataset therefore seems to be a bit easier for horizon line-based methods, our keypoint-based method performs similarly on both GeoPose3K and CH1 dataset in absolute numbers.

Finally, we use the Venturi Mountain dataset [142] to compare our keypoint-based method with the Confidence Fusion method (CF-VCC-2011) combined with the edge-based non-linear metric (CF-VCC-2011-m3D)—both presented in Chap. 6, original edge-based non-linear metric VCC-2011-m3D (proposed by Baboud *et al.* [15]), and the horizon-line based HLoc method. Same as with other methods, we evaluate our keypoint-based method independently frame-by-frame without any constraints in the temporal domain. The results are shown in the Tab. 7.4.

Our keypoint-based method fails (at least partly) on sequences F2, F3, F5, J3, and J5. The failure is caused by a predominant high-frequency foreground forcing the used keypoint detector (SIFT) to detect non-descriptive keypoints. Another difficulty is a low resolution of the input frames, which is 640x480 px—we designed our method for input images of almost double resolution, 1024x768 px. We believe that the failure cases could be compensated by a more specialized keypoint detector, which would discard non-informative foreground keypoints and detect more descriptive keypoints on more distant (and therefore more stable) objects. On the other hand, our method delivers the best results on sequences F1, F4, J1, and J2 (see the first line in Tab. 7.4). In these sequences, there are enough keypoint features to be matched easily. This result illustrates that for inputs with distinctive and descriptive keypoint features, our method can easily surpass methods based on edges or semantic segments, even though it estimates not only the camera orientation but also the position.

7.4 Qualitative Evaluation

We illustrate several qualitative results of our method in Fig. 7.10. In the top row, we see that our keypoint-based approach, unlike the horizon line-based methods [163], can precisely estimate camera pose even for images where no horizon line is visible. Additionally, we expect our approach to work well if around 100 inliers distributed all over the photograph are available.

We found that our approach most likely fails on images fully covered by snow (see the top row of Fig. 7.11), containing lots of high-frequency noise in the foreground (usually caused by foliage or trees), or when the photograph contains mostly flat terrain (see the bottom row of Fig. 7.11), where the amount of overlapping keypoints with the rendered image is low.

7.5 Applications

Mobile Application. To demonstrate the practicality of our method, we implemented it in an iPhone application. The application takes a camera stream, an initial rotation, and position derived from on-board device sensors, and renders synthetic views from the local DEM and orthophoto textures. It then computes SIFT keypoints on both a still image from the camera stream and the synthetically rendered image and uses our trained

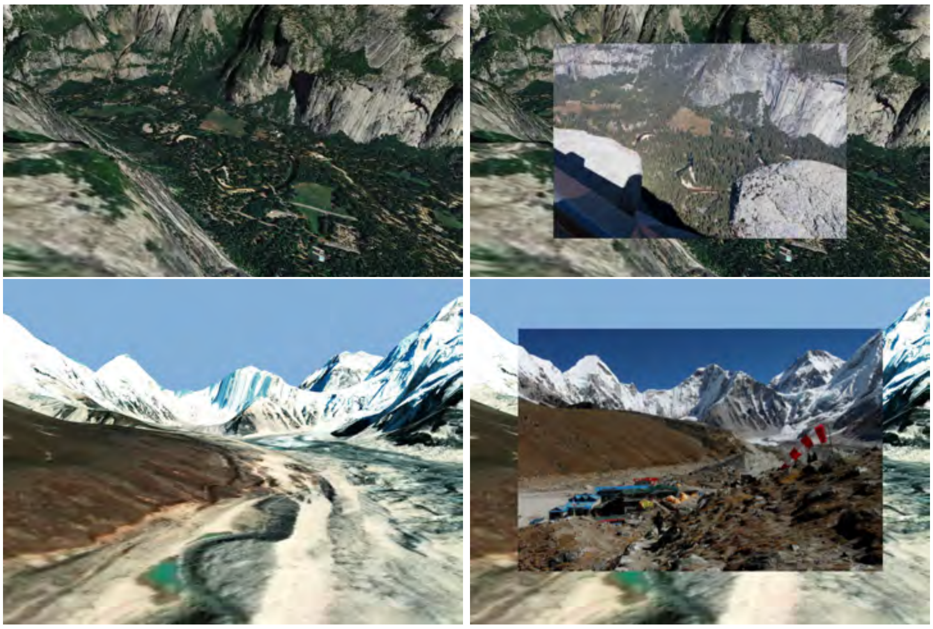


Figure 7.10: Illustration of successful results obtained by pose estimation using our cross-domain matching method. Left: terrain rendered with the estimated camera postion and rotation, right: the rendered image overlaid by the photograph. First line: Yosemite Valley (image credit Kirk Northrop, https://flic.kr/p/22MAjoC), second line: Nepal—view from Gorakshep.

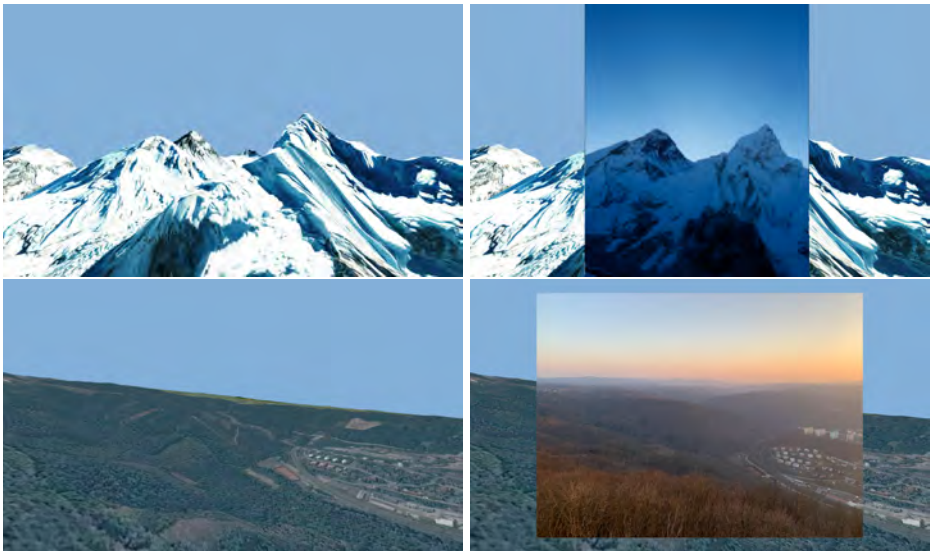


Figure 7.11: Illustration of inaccurate results. Left: terrain rendered with the estimated camera postion and rotation, right: the rendered image overlaid by the photograph. First line: Mount Everest and Nuptse, second line: view from Alexandrovka—an observation tower near Babice nad Svitavou, Czech Republic.

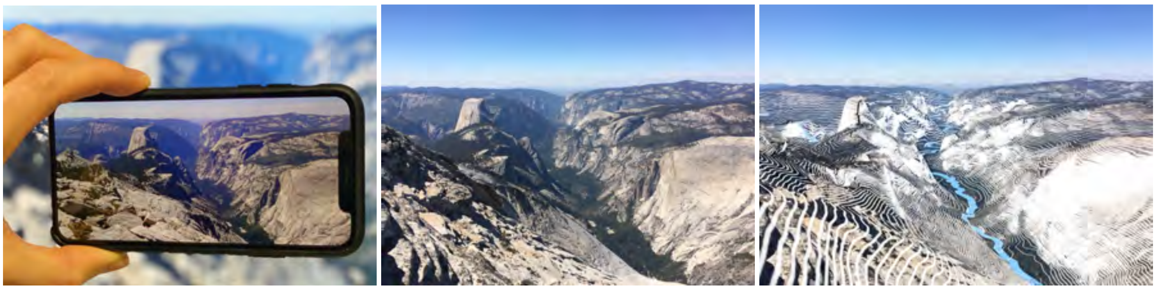


Figure 7.12: An iPhone application (in the left) is used to capture the photograph (in the middle) for which precise camera pose is estimated using our method. The estimated camera pose (in the right) is used to augment the query photograph with contour lines (white) and rivers (blue).

CNN to extract local features on the detected keypoints. These features are matched across domains and are then unprojected from the rendered image using the camera parameters and the depth map. Finally, matches between the 2D still keypoints and 3D rendered keypoints are used to estimate the camera pose using the PnP method with RANSAC. This estimated camera pose is used to update the camera position and rotation to improve the alignment of the input camera stream with the terrain model (see Fig. 7.12).

Automatic Photo Augmentation. Furthermore, we demonstrate another use-case of our camera pose estimation approach by augmenting pictures from the internet for which the prior orientation is unknown and GPS position imprecise (see Fig. 7.10). Please note that many further applications of our method are possible, *e.g.*, image annotation [99, 15], dehazing, relighting [99], or refocusing and depth-of-field simulation [30].

7.6 Chapter Summary

We have presented a method for photo-to-terrain alignment for use in augmented reality applications. By training a network on a cross-domain feature embedding, we bridged the domain gap between rendered and real images. This embedding allows accurate alignment of a photo or camera view to the terrain for mobile augmented reality (AR) and photo augmentation applications.

Our approach compares favorably to the state-of-art in alignment accuracy and is much smaller and more performant, facilitating mobile applications. We see this method as especially applicable when virtual information is to be visually aligned with real terrain, *e.g.*, for educational purposes in scenarios where sensor data is not sufficiently accurate for the purpose. Looking forward, we expect that our method could be made more performant and robust by developing a dedicated keypoint detector capable of judging which real and synthetic points are more likely to map across the domain gap.

Chapter 8

Immersive Trip Reports

The human desire to travel dates back to before written history. So does, it would seem, the desire of travelers to share the experiences from their journeys. Travel literature is known to us since antiquity, and was a staple of medieval and early modern writing [187, 137, 138]. More recently, as photography became widespread, it started to be widely used to record and share impressions from travels and vacations, indicating a desire to convey these experiences in a more engaging and immersive way.

Previous research has explored putting the photographs in a spatial context by manually registering them to a topographic map represented as a DEM through tools such as PhotoOverlay in Google Earth [38]. Photo un-cropping methods [175, 234] mine collections of external photographs for visual data to extend the field-of-view of the user's own photos. Structure from motion (SfM) methods register large collections of photographs of an artifact to create a 3D model, allowing a structured exploration of the photo collection [184, 185, 183, 104]. An extension of a structured exploration based on a SfM reconstruction uses accurate 3D models of urban environments to align the reconstructed scene and photographs with the physical geometry [208].

In this chapter, we utilize recent advances in computer vision and virtual reality to increase the immersiveness of a photo presentation. Specifically, we have developed a process, illustrated in Figure 8.1, to extract 3D location and orientation information from collections of photographs taken on hiking trips, which we further use to align the photographs to a virtual representation of the actual terrain. We use this information to enrich the presentation with supplementary geographic data and replay the experience from a first-person perspective. We show that this pipeline works in general landscapes and requires only rough DEM data. By using the recovered information to automatically place the photos in the virtual terrain, we facilitate a rich first-person exploration experience that supplements the aesthetic and informational value of the photographs with contextualized spatial information.

We might divide our method's target audience into two groups: (1) hikers who wish to share the experience of a hike, and (2) viewers who wish to learn more about hikes at locations they have not yet visited. Users from the second group who enjoy the presen-

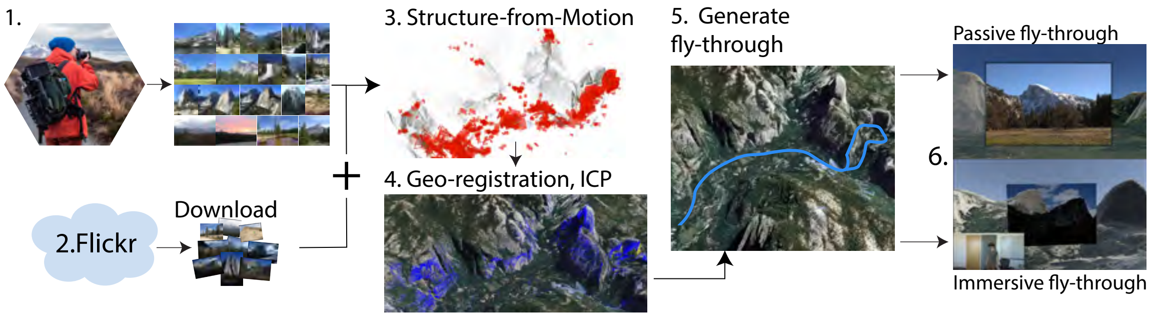


Figure 8.1: Our virtual trip creation pipeline: 1. User takes photographs during a hike; 2. We augment the input collection with images downloaded from Flickr.com; 3. Camera positions and sparse 3D point cloud reconstruction using *Structure from Motion*; 4. Scene alignment with the terrain using ICP; 5. Fly-through generation from the input photographs from the hike; 6. We export the fly-through to Google Earth or to our virtual reality viewer. Map data © 2018 Google, © Mapbox, © OpenStreetMap.

tation may then re-create the hike themselves. Therefore, the purpose of sharing travel photographs is not just to enjoy the scenery, but to convey the entire experience of visiting the remote location.

Our goal is that our enhanced photo presentation will assist viewers to gain spatial orientation, better understand the scene, and enjoy the viewing experience. To evaluate these effects, we conduct a user study comparing four different modes of presentation (illustrated in Figure 8.2) on four datasets from different locations. The tested modes consist of a traditional slideshow, a slideshow with GPS markers shown on a map (GPS slideshow), and two modes produced by our method. A fly-through from photo to photo precisely aligned with a virtual terrain model was in one mode viewed passively as a rendered video (passive fly-through), and in the other interactively in virtual reality (interactive fly-through).

Contributions. We automatically generate new modes of immersive first-person presentation of photographs, specifically a passive fly-through, renderable as video and compatible with tools like Google Earth, and an interactive fly-through which presents the trip in virtual reality. We also conducted experiments demonstrating that these immersive presentation modes help users understand the spatial relations in the region significantly better than a traditional slideshow and that the interactive virtual reality (VR) experience is enjoyable.

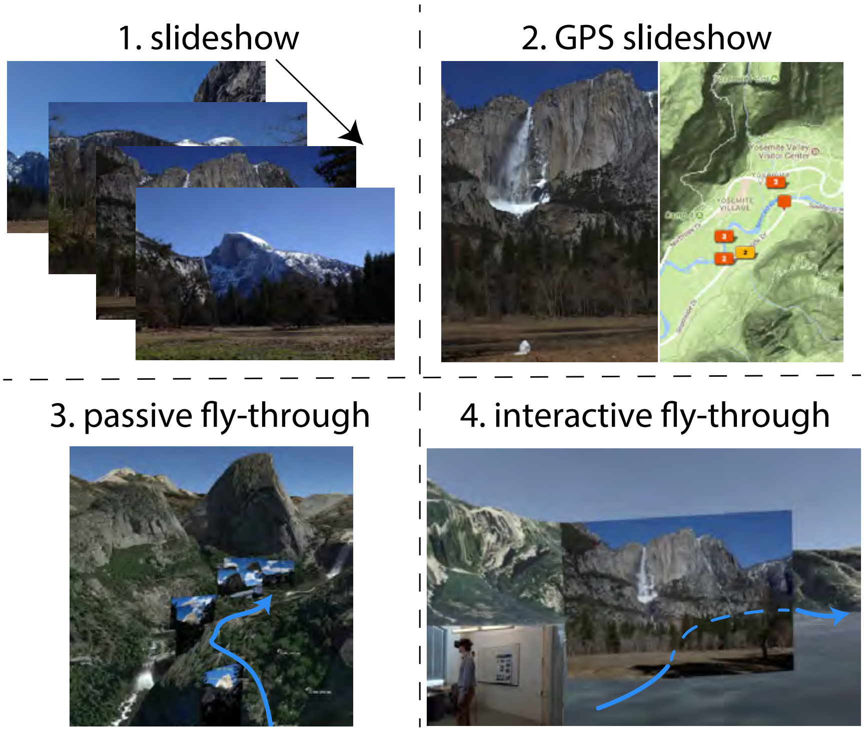


Figure 8.2: Visualization of four modes of presentation. 1. slideshow: photographs are presented sequentially. 2. GPS slideshow: the slideshow with a map showing the position of currently shown photograph. 3. passive fly-through (ours): photographs aligned with the terrain are presented in a passive fly-through. 4. interactive fly-through (ours): the user can freely look around during the fly-through. Map data © 2018 Google, © Mapbox, © OpenStreetMap.

8.1 Related Work

8.1.1 Photography Presentation

Previous research has explored alternative presentations of photographs, often based on 3D scene reconstruction using SfM [184, 183, 104, 182]. However, 3D point clouds used by PhotoTourism and others [104, 183, 184] are not suitable for visualization of a re-created trip. For example, in natural environments, usually only front facing parts of mountains are reconstructed leading to incomplete point clouds. Since a tour can traverse widely spaced viewpoints, the partial model reconstruction may result in poor visuals between photographs. Our method solves this problem by using the terrain model, which is more suitable for presentation of the whole trip.

Visualization of images with geographical information is available commercially via online services such as Flickr and Google Maps. Researchers have explored visualizing photographs in a map online [201] or in virtual reality [135]. Geo-tagged social media enables spatial navigation interfaces for photo albums [198], even composited atop panoramas from Google Street View [47]. Note that these interfaces are not designed to convey a virtual hike experience. VR BBS [135] is for sharing photographs and messages in a virtual environment, with users plotting their course through a flat map with 3D sprites of photographs. In contrast, our system leads the user automatically through virtual terrain containing the sequence of photographs of a re-created trip. Additionally, these previous works do not precisely align image content with the environment. Precisely aligned image with the virtual environment is vital in the seamless in-situ visualization implemented in our method.

The work most similar to ours is Kuchelmeister *et al.*'s [103] presentation of an immersive visualization of photographs taken by SenseCam jointly with a virtual model of a 3D outdoor scene. Their work intends to study the effect of browsing photographs in this virtual environment as a memory-prosthesis for patients suffering from amnesia. In contrast, our work does not use any specific device for collecting photographs, and our experiments are focused on the orientation of users in the presented space and enjoyment of such a visualization.

In summary, previous methods are not designed to re-create a virtual hike experience. Specifically, we focus on the single-user-multiple-landmarks scenario, whereas PhotoTourism [184] addresses the multiple-user-single-landmark scenario. The selection of the single-user-multiple-landmarks scenario has algorithm implications, so *e.g.*, PhotoTourism and VR BBS [135] require much more elaborate capture processes. Our key idea is to download additional imagery to help the reconstruction (see Figure 8.1), but use only user-generated photographs for the presentation.

8.1.2 Photography Management and Categorization

Our immersive presentation is related to photo browsing and management systems. The rapidly growing number of photographs being taken has motivated research into effective searching [91] and clustering of photographs [139, 140], which can also be based on space and time [63, 219]. The difficulty of browsing, sorting, and clustering photographs manually has led to novel interfaces such as Photohelix [81]. Rodden and Wood [154] show that users tend to use simple features of photo management software, and also that managing photographs digitally is easier than managing printed photos. Harada *et al.* [71] designed an automatic searching and browsing tool for photographs on mobile devices. Schoeffmann *et al.* [167] show that photographs organized into a 3D cylinder or globe help users with faster visual search.

8.1.3 Related Applications

Researchers have explored narrative storytelling with mobile photos [16] or photo blogs [96], or even writing fictional stories [147], as alternate ways of facilitating user engagement. Chelaramani *et al*. used photos of a historical site to create a multimedia tour guide for cultural heritage [33]. Another work for cultural heritage has combined photos with animations [182]. Immersive presentations such as virtual reality [76] and mobile augmented reality [75] have been found to improve appreciation of historical sites [44].

For productivity applications, PhotoScope [219] combines photo albums and building floor plans to aid construction management. Immersive presentations of many video feeds have been used to support video surveillance tasks, with desktop spatial navigation [66], desktop 3D environments [171], or full immersive virtual reality presentation [46]. Taken together, these related applications all support the notion that presenting photo albums of a remote location in virtual reality can improve users' engagement with the presentation and their resulting understanding of the experience.

8.2 Automatic Generation of Photography Presentations

The pipeline we use in our method is visually summarized in Figure 8.1; a more detailed flowchart of the pipeline can be found in Figure 8.3. Our goal is to reconstruct from photographs a real hike in a virtual model of the real terrain. The input to our method is a collection of photographs taken on a hike I_h, together with the geo-rectangle designating their rough geographical extent, which can be read from embedded GPS information if available. We take the user photographs I_h as-is, we do not consider color enhancement as a part of our pipeline. We augment this collection with additional photographs from the same geo-extent I_f, which can be harvested from online repositories such as Flickr, to improve coverage of the terrain for better reconstruction. We jointly mine the merged photoset for both GPS metadata and visual features, which we use to obtain a rough georegistration through a SfM pipeline. We align the result of the reconstruction with known DEM terrain data to fine-tune the camera estimation. Finally, we construct a virtual presentation that shows selected photographs and renders fly-throughs from one camera pose to the next as a transition between the consecutive photographs.

8.2.1 Imageset Augmentation and Scene Reconstruction

We conducted initial experiments with datasets from the authors' personal collections in a variety of locations. Although these datasets were uncurated (*i.e.*, contained all the photographs taken, including those that would not be selected for presentation), we found

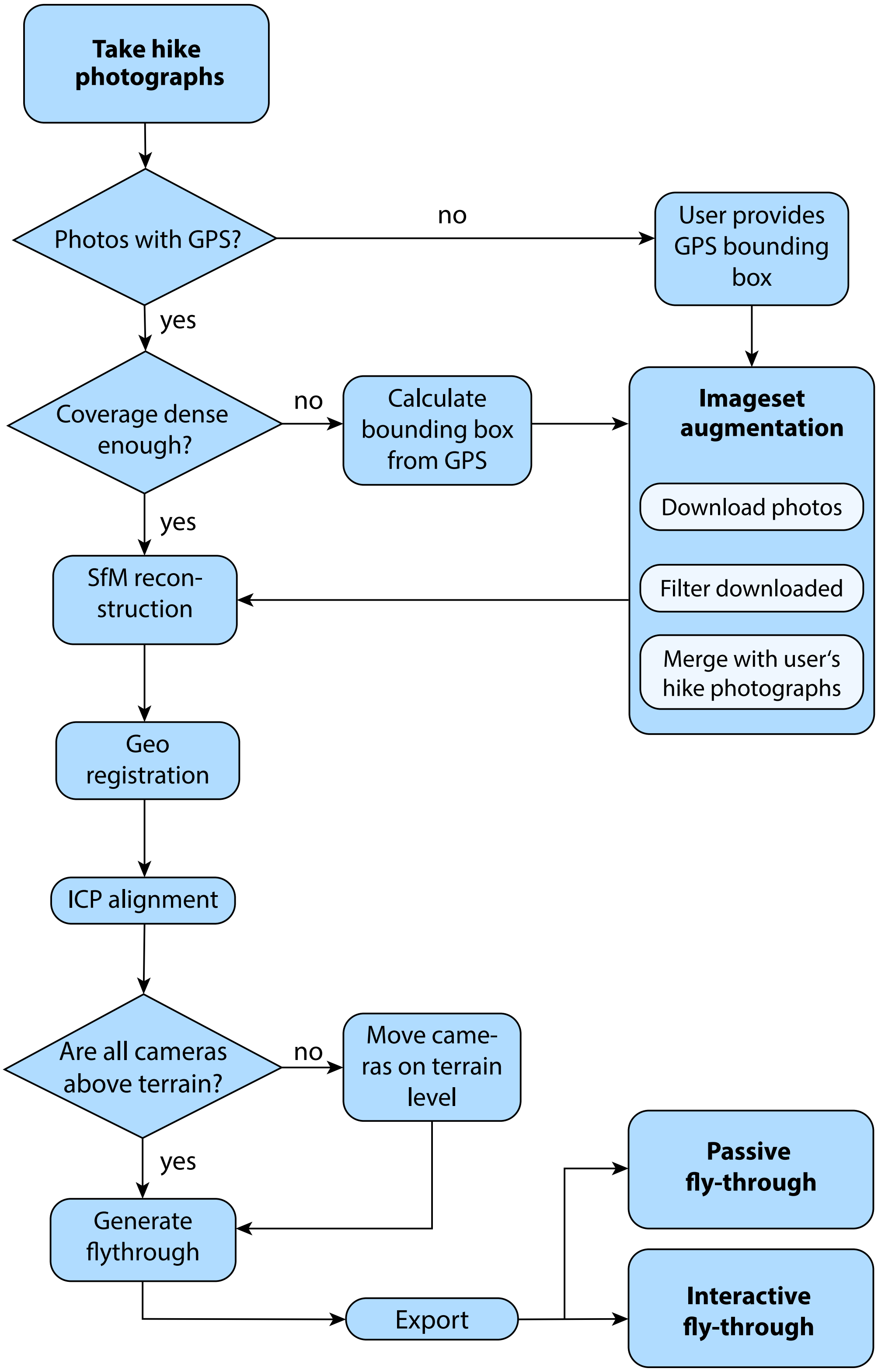


Figure 8.3: Flowchart illustrating complete process of our virtual trip creation pipeline.

that a single user does not usually provide sufficient coverage of the space for a reliable 3D reconstruction. Coverage density may be tested by running the matching stage of the SfM reconstruction on the set of user photographs I_h. If the number of matching images with strong matches is low, we perform *imageset augmentation*. We augment each of the original collections I_h with images downloaded from Flickr I_f. The augmentation has the additional advantage that the original dataset I_h need not contain GPS information, since we may use GPS from the downloaded photographs I_f. However, in the absence of any GPS information in user photos I_h, we need the user to provide the rough extent of the visited area, specified as *e.g.*, center and radius. The detailed description of downloading the Flickr photographs I_f, processing, and reconstruction of the mixed collection $I_m = I_h \cup I_f$ is described in Section 5.2.

8.2.2 Fly-through Creation

For the fly-through presentation, the user selects a curated subset of photographs $I_c \subseteq I_h$ based on their aesthetic preference. Although we know the camera pose for each photograph from the registration, we still need to estimate the actual hiking path from one camera position to the next. We generate a smooth camera path by constructing a Catmull-Rom spline with the camera positions from I_h as control points. Alternatively, if a full GPS track is available, it may be used as the camera path instead, to ensure that the presentation follows the trail between photographs. Please note that the selection of the curated subset of photographs I_c only affects *which* photographs will be presented; the reconstructed path is the same for different subsets of I_c.

We initialize the set of control points P_c with the positions of the curated photographs I_c. We add the remaining positions from the reconstructed photos I_h in a greedy way—we add a point only if it is further than 100 m from all points in P_c. We sort the control points P_c according to the time of capture of the corresponding photograph parsed from EXIF. We generate the Catmull-Rom spline from the selected control points P_c. In case any point of the spline is located below the terrain, we project it above the terrain level by a fixed margin. We smooth the generated spline using a low-pass box filter.

A part of the spline between consecutive control points is called a *segment*. In passive mode, as the camera moves along a segment, we smoothly interpolate camera parameters. Field of view is interpolated linearly between photographs of consecutive control points; the camera orientation is interpolated to look in the direction of the next control point. For transitions from one photo to the next, we use spherical interpolation between the two orientations, with the camera located at the center of the sphere to achieve near-constant angular speed. We calculate the speed of the camera automatically—for more distant control points the camera flies faster, accelerating and decelerating at the start/end of the segment, respectively. In the interactive mode, the field of view and orientation are defined by the output device (*e.g.*, the headset), and the speed of the flight is controlled directly by the user. Also, in interactive mode, the user can move in a small neighborhood of the current position on the spline. In passive mode, the position of the camera is restricted to the generated spline.

We combine the fly-through with the photographs rendered with appropriate camera parameters over the virtual landscape to generate the actual presentation. In the passive case, we cross-fade from the end of a fly-through segment to the photo we wish to display and then cross-fade to the next segment. In the interactive mode, the cross-fade for leaving the photograph is triggered by the user. In both cases, accurate estimation of camera parameters ensures the transitions are smooth.

8.3 Experiments

The goal of our method is to create an enjoyable presentation that helps the viewer understand the physical layout of the place where the photographs were taken. We conducted a user study that compares four modes of presentation of photographs; two traditional and two based on our method. We evaluate these methods on viewer enjoyment, sense of presence, and a quantitative task that measures how well the user can localize previously unseen photos from the same space after viewing the presentation. First, measuring enjoyment is important to understand if users want to use our method. Second, we measure the sense of presence to determine how immersed users become on a virtual hiking trip. Third, we assess users' orienteering capability conditioned on the presentation method to determine if our method measurably impacts users' spatial understanding of the environment. We use four datasets processed with our pipeline, and from each dataset, we select one subset of photographs for presentation and a disjoint subset for evaluation.

8.3.1 Datasets

Out of the four datasets we used in our experiment, three were captured manually at the Lake Tahoe, CA, USA, Yosemite Valley, CA, USA, and the Himalaya mountains in Sagarmatha National Park, Nepal. The fourth dataset from the High Tatra mountains in Slovakia was collected from Flickr. Each dataset was captured by a different photographer. The Lake Tahoe dataset was reconstructed directly without any additional photographs, while Yosemite and Nepal were augmented using Flickr images. The statistics on the number of captured photographs I_h, photographs downloaded from Flickr I_f, successfully registered user photographs I_{hr}, and total successfully registered photographs I_{mr} are shown in Table 8.1. All four datasets were processed with our geo-registration pipeline and ex-

dataset	I_h	I_f	I_m	I_{hr}	I_{mr}
Nepal	1586	815	2401	412	901
Tahoe	302 (36)	0	302	78 (7)	78 (7)
Tatras	0	4146	4146	297	297
Yosemite	543 (117)	4173	4716	167 (33)	2094

Table 8.1: Number of photographs in our datasets. I_h—input hike photographs captured by user, I_f—number of downloaded Flickr images, I_m—number of mixed photographs entering the reconstruction, I_{hr}—number of hike photographs that were successfully reconstructed, I_{mr}—number of all reconstructed photographs. Panoramic images are included and denoted by numbers in brackets.

ported to Google Earth through KML for the passive mode and to our implementation of a VR viewer in Unity with the terrain loaded from Mapbox for the interactive mode.

8.3.2 Modes and Setup

For evaluation, we use four datasets created by the reconstruct-then-align approach described in Sec. 5.2. We compared four modes of presentation, shown in Figure 8.2. The baseline mode, *slideshow*, is a standard photo slideshow without any additional information. The second mode, *GPS slideshow*, is a slideshow with camera positions marked on a map presented in Adobe Lightroom. For each photograph, the user can explore a Google "terrain" map with contour lines in a fixed zoom level, where all the photographs in the presentation are localized, and the current one is highlighted. The third mode, *passive fly-through*, is the passive version of our method: a fly-through in Google Earth generated by our geo-registration pipeline. The user is first shown the path of the tour in a top-down view. The view then transitions to the camera position and orientation of the first photograph, with the photograph drawn over the terrain. As the user presses a button, the view flies to the next camera position and shows the next photograph in the same fashion. Once the fly-through is finished, the presentation returns to the initial top-down view. The final mode, *interactive fly-through*, is the interactive version of our method, with the fly-through presented in VR. The user is first allowed to familiarize themselves with the region's terrain from a bird's-eye view several kilometers up. They are next teleported to the fly-through, which proceeds in a similar fashion to the passive mode, except the user has the opportunity to look around freely and can control the speed of movement along the camera path in order to reduce the risk of motion sickness.

dataset	positional error	heading error
Tahoe discr.	0/6	0/6
Tahoe cont.	353.61 ± 230.29m	32.05 ± 28.39°
Yosemite discr.	0/4	3/4
Yosemite cont.	1189.33 ± 748.61m	23.81 ± 20.44°
Nepal cont.	4710.74 ± 2833.38m	75.14 ± 53.34°

Table 8.2: Pilot study data. For discrete version the numbers denote a fraction of wrong answers. For continuous measurements the mean and standard deviation is reported.

In all modes, the user sees each photo only once without the option to go back. All modes and datasets were presented on a calibrated[1] 15″ MacBook Pro Retina display in native resolution 2880 × 1800 pixels under office lighting, except the *interactive fly-through* mode, which was presented using an HTC Vive. Each participant tested all four modes, each with a different dataset to avoid learning effects. The mode-dataset pairing and the mode order were randomized for each participant.

8.3.3 Pilot Study

To help design the main study, we performed an initial experiment with one participant. The female participant was a co-author of the *Tahoe* dataset and familiar with the terrain in the *Yosemite* dataset, with extensive experience in using maps for navigation. The purpose of this test was to determine whether the task is better evaluated using discrete or continuous questions. The participant was first shown a presentation of at most 20 photographs and afterward was asked to complete a task with a selection of photographs from the same dataset but disjoint from that shown in the presentation.

In the discrete scenario, the participant answered binary questions about camera heading and position. For position, she was shown a query photograph taken chronologically between two consecutive photographs from the presentation, and asked to identify whether the viewpoint of the novel photograph is closer to that of the earlier photograph or that of the later photograph. For heading, she was shown a query photograph and a reference photograph from the presentation and asked to identify whether the query photograph camera orientation is to the left or right of that of the reference photograph.

In the continuous scenario, we asked the participant to mark two points in an online map for each photograph. The first corresponds to the camera viewpoint of the query photograph. For the second, the participant could pick an arbitrary reference point in the

[1] The calibration was performed by X-Rite GretagMacbeth Eye-One Display colorimeter to D65, 120 cd/m^2, and colorimetrically characterized by measured ICC profiles.

query photograph and then select a point on the map that corresponds to the location marked in the photograph (see Figure 8.4).

Initially, we tested the *Tahoe* dataset in the *slideshow* mode, and the *Yosemite* dataset in the *passive fly-through* with both discrete and continuous variants. For each variant, we tested 4–6 different photographs. Since the participant visited both areas earlier, we added a test on the *Nepal* dataset in a *passive fly-through* with a continuous variant. We show the results in Table 8.2. The discrete and continuous variants are consistent on the *Yosemite* dataset; the participant could estimate heading more accurately than the position for both task sets on this dataset. Conversely, even though the participant achieved perfect success on the discrete heading task for both *Tahoe* and *Yosemite*, the continuous heading error is higher on the former. The continuous errors are notably higher on the *Nepal* dataset, suggesting a significant difference in difficulty between datasets, possibly related to the spatial extent and complexity of the terrain. The participant expressed her preference for the continuous tasks, describing them as an exciting puzzle instead of the discrete tasks that she tended to answer randomly when in doubt. Another issue in the discrete task is that when the rotation is close to 180° compared to the reference, it is extremely difficult for the participant to correctly answer, as the difference between "left" and "right" is only a few degrees. Based on these observations, we selected the continuous task set as the evaluation method for the full user study. We expected it to give us more information with less variance, even with a small number of participants, which was limited by each test's long duration (up to 1 hour for all four modes with each participant). We also expected the continuous task set to be more engaging for the users and thus keep them more focused. Finally, we realized the necessity of normalizing errors per-dataset due to high observed variation in dataset difficulty.

We also performed a field-type experiment where we presented photographs from the *Nepal* dataset in the *slideshow* and the *passive fly-through* presentation modes to a broader audience of approximately 40 people. After the presentation, the audience completed a short questionnaire asking which of the two methods they preferred more and whether the terrain model helped them better understand the positions and orientations of the photographs compared to the slideshow. Out of 40 participants, 22 completed the questionnaire. Regarding the first question, 8 participants replied that they liked the fly-through more than the slideshow, 10 participants liked both roughly the same, and 4 participants liked the slideshow mode more. Responses for the second question were even more optimistic: 14 participants agreed that the fly-through helped them, 2 participants replied that both modes helped them roughly the same, and 4 participants replied that the slideshow helped them more. A final question asked participants to write what they liked or disliked. Participants disliked the abrupt speed of camera rotations during transitions in the

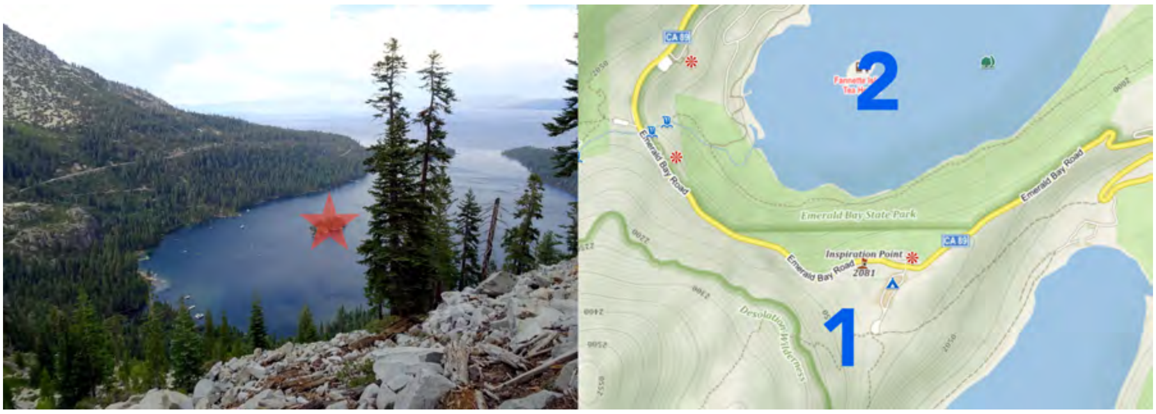


Figure 8.4: Example task from Lake Tahoe. The participant marks a position on the map (right image #1) of the query photo (left image) and the reference point (left image, red star) and corresponding position of the reference point on the map (right image #2). Map data © Mapy.cz.

fly-through. We identified this as the main reason why 14 out of 22 participants preferred the slideshow or had no preference in the first question. Due to this finding, we adjusted the angular velocity to ensure smooth camera rotations. Furthermore, subsequent experiments were designed based on the experience from this field experiment.

8.3.4 Evaluation Methodology

Each participant was instructed about the purpose of the experiment and completed a screener questionnaire. Before the first experiment, we explained the task with a dummy example. The procedure was as follows: a presentation of at most 20 photographs was shown to the participant. The participant viewed one picture at a time, as determined by the presentation mode. The participant was not allowed to return to previously viewed photographs. After the presentation, the participant viewed 6–7 photographs not present in the presentation but taken on the same dataset. For each photograph, we performed the continuous task variant, as determined by the pilot study, in which the participant indicated the position and heading of the camera by marking a map. The participant was not allowed to move already placed marks once they continued to the next photo, or to return to a previous photo during the test. The participant was allowed to zoom in to the online map during the task, and to move around within the area of the dataset. If they moved out of the area, the moderator would reset the map to the initial view. The initial zoom level was chosen so that the area of the whole dataset would fit inside the window. The digital map featured a top-down view with only the names of points of interest (POI), tourist pathways, and contour lines showing elevation.

8.3.5 User Study

Participants. We assembled 21 volunteers, predominantly bachelor and master students of informatics (17) and law (4); 3 women and 18 men. One participant had been to Lake Tahoe, 4 to Yosemite, 13 to High Tatra Mountains, and 1 to Nepal. Fourteen participants had some experience with virtual reality. Each participant had at least basic knowledge on how to use a map: one participant used maps several times in his life, 5 participants used maps at least once a year, 10 participants used maps at least once a month, and 5 used maps at least once a week. Where possible, we correct our experimental data for the bias introduced by these factors.

Error measures. We report two error measures per test photograph: the positional error e_p (Eq. 3.16), and the heading error e_h. The heading error is the smallest absolute difference between the ground truth heading $h_g \in [0, 360]$ and the measured heading $h_m \in [0, 360]$ (in degrees): $e_h = \min(|h_g - h_m|, 360 - |h_g - h_m|)$.

Positional error model. We use different datasets for each test to avoid learning effects, but this introduces the possibility that performance may be correlated with dataset difficulty. To compensate for dataset and user differences, we model the positional error as a normal random variable $e_p \sim \mathcal{N}(sdm, \sigma^2)$, where s is a factor of the subject's ability, d is a factor of the dataset difficulty, m is a factor of the mode properties, and σ^2 models measurement noise. Since we want to compare modes based on the positional error, we need to mitigate the effects of dataset factor d and subject's ability factor s.

We expect that *Nepal* and *Tatras* are more difficult than *Tahoe* and *Yosemite* because the trips made in *Nepal* and *Tatras* are much longer, and the terrain profile is more complicated. Figure 8.5(left) confirms this, but the positional error e_p has a different scale for each dataset due to different geographic extents. One-way ANOVA clearly rejected the null hypothesis ($F(3, 542) = 149.85, p < 0.001$), that the means of positional errors e_p do not vary significantly across datasets. Further inspection reveals that the *Nepal* dataset has significantly higher positional error than other datasets across all methods. We attempted to normalize the errors by dividing it by the dataset extent. This normalization moved the scale between datasets closer, but the null hypothesis was still clearly rejected ($F(3, 542) = 10.85, p < 0.001$). In this case, the *Tahoe* dataset was shown to have a significantly lower mean error than other datasets. Instead, we use the baseline mode *slideshow* as a dataset calibration measure. We calculate the normalized positional error $e_{np}(d)$ for each dataset d by dividing by the mean of the positional error $e_p(d, m_s)$ for the *slideshow*

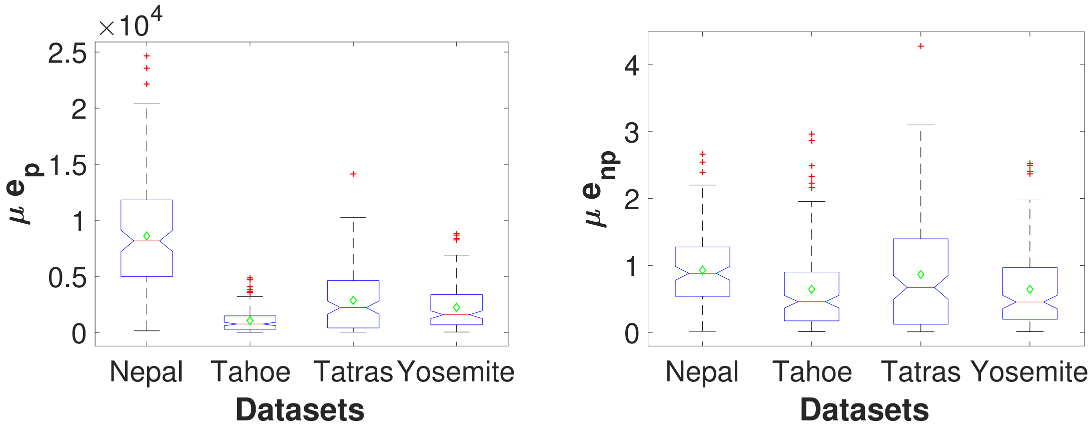


Figure 8.5: Differences between datasets before normalization of mean positional error μe_p (left) and after normalization μe_{np} (right). The central red mark indicates the median, the green diamond denotes the mean, the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme positional errors not considered outliers, and the outliers are plotted individually using the '+' symbol

mode m_s and the dataset d:

$$e_{np}(d) = \frac{N_d e_p(d)}{\sum e_p(d, m_s)}, \tag{8.1}$$

where N_d is the number of measurements for dataset d. This yields the lowest F-score compared to other normalization methods ($F(3, 542) = 6.98, p = 0.0001$). The null hypothesis is still rejected, due to the fact that the baseline *slideshow* mode has been tested by different users on different datasets. However, the error distributions have almost the same scale, and the result still matches our initial expectations: the *Tahoe* and *Yosemite* datasets exhibit lower mean error than the *Nepal* and *Tatras* (see Figure 8.5 right).

Subject's ability factor. We tested the per-subject mean differences using one-way ANOVA. The test was unable to reject the null hypothesis that the means of positional error e_p do not vary significantly between users ($F(20, 525) = 1.13, p = 0.31$), which also hold for the normalized positional error $e_{np}(F(20, 525) = 1.75, p = 0.23)$. We further inspected the importance of factors that the subject visited the place before, map proficiency, and map usage frequency. None of them showed a significant effect on positional or heading error. In summary, we could not prove any significant differences between users in terms of positional and heading error.

Position evaluation. Having normalized results for dataset difficulty, we can formulate the comparison of presentation modes as one-way repeated measures ANOVA with the presentation mode as a within-subject variable with four conditions. This way, the test can

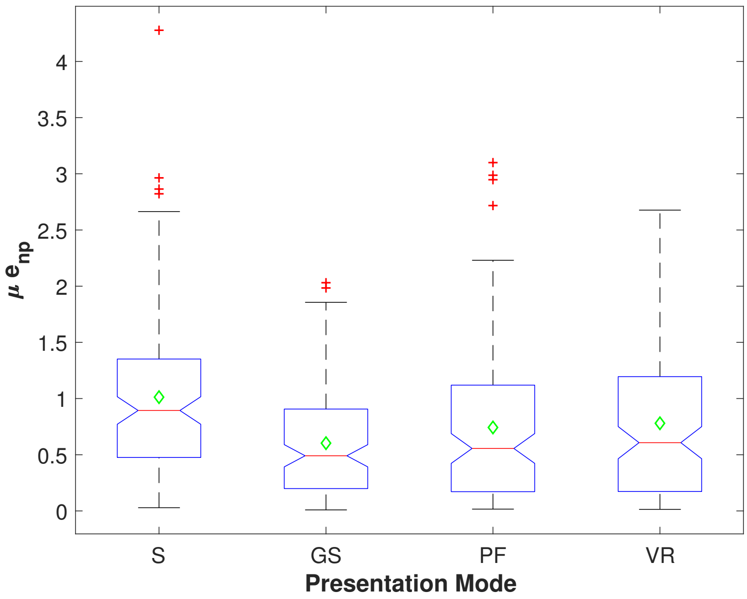


Figure 8.6: Repeated measures scenario comparing differences between normalized positional error e_{np} on different modes of presentation (S = *slideshow*, GS = *GPS slideshow*, PF = *passive fly-through*, VR = *immersive fly-through*). The mean value for each method is denoted by green diamond.

account for performance differences between subjects. As the numbers of photographs differ between datasets, we first calculate a mean per-subject and method. This way, we have one measurement per subject and method. We formulate the null hypothesis that means of normalized positional error e_{np} do not differ significantly between methods. The null hypothesis was clearly rejected ($F(3, 375) = 8.13, p < 0.001$). Post-hoc analysis reveals that the baseline presentation mode has significantly larger mean normalized positional error e_{np}, than *GPS slideshow* ($p < 0.001$), *interactive fly-through* ($p = 0.034$), and *passive fly-through* modes ($p = 0.009$, see Figure 8.6). There is no significant difference between the *GPS slideshow*, *passive fly-through*, and *interactive fly-through* according to our data and this test ($p >= 0.434$) for all remaining combinations). In summary, it seems the positional information contained in *GPS slideshow*, *interactive fly-through*, and *passive fly-through* modes help users with location estimation.

Heading evaluation. We were not able to find any significant differences between presentation modes for heading error e_h. We found a significant difference between the *Tahoe* and *Nepal* datasets using one-way ANOVA ($F(3, 542) = 4.23, p = 0.0057$), supporting our expectation that the *Tahoe* dataset is easier than *Nepal* (and according to Figure 8.7 left probably the easiest among all datasets). Our data suggest it is fairly difficult to under-

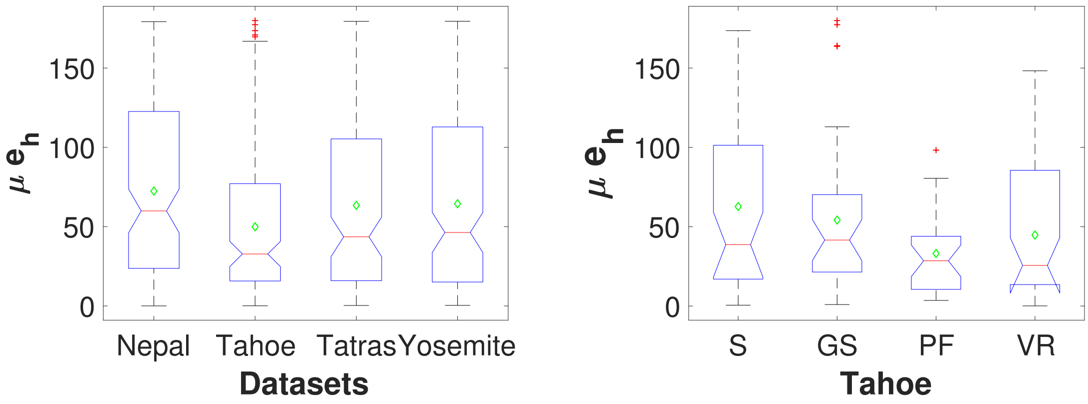


Figure 8.7: Left: comparison of dataset difficulty with respect to heading error e_h. Right: comparison of heading errors achieved by presentation modes (S = *slideshow*, GS = *GPS slideshow*, PF = *passive fly-through*, VR = *immersive fly-through*) on the easiest *Tahoe* dataset.

stand what the camera is looking at in a photo and then mark it on a map. The only dataset where the orientation exhibits some tendency is the easiest *Tahoe* dataset. The *passive fly-through* has the lowest mean heading error, and *interactive fly-through* has the second-lowest (see Figure 8.7 right); however, these differences are not statistically significant. Other datasets seemed to be too difficult for heading estimation as all the methods exhibited similar variance and mean across the remaining datasets, probably due to large random error. In summary, on the easiest *Tahoe* dataset, the *passive fly-through* and the *interactive fly-through* seem to have marginally lower orientation error than the remaining two presentation modes.

Presence evaluation. We included a presence questionnaire to evaluate how successfully the user is immersed by each presentation mode. To reduce the time of the experiment, we tested just two modes of presentation—the *GPS slideshow* and the *interactive fly-through* on a randomly selected half of our participants. For this evaluation, we use the SUS presence questionnaire [181] because of its relative compactness. As a first measure, we calculate the number of high responses (6, 7) for each presentation mode—7 for *interactive fly-through*—and 6 for *GPS slideshow* (higher is better). We also calculate the mean and standard deviation of scores for both methods: the *interactive fly-through* is 3.94 ± 1.40, and the *GPS slideshow* is 3.29 ± 1.68. We can see that the *interactive fly-through* is better than the *GPS slideshow*; however, one-way ANOVA does not find significance. In summary, the *interactive fly-through* seems to exhibit slightly better scores in terms of presence compared to the *GPS slideshow*.

In the post-test questionnaire, we asked users whether they think that the terrain model (*passive fly-through* or *interactive fly-through* modes) helped them create a better idea about the dataset area. The terrain model was helpful for 7 participants, 8 participants thought the terrain model helped them roughly the same as the *GPS slideshow*, and 6 participants replied that the *GPS slideshow* helped them more.

Enjoyment. We asked users to identify which method was the most enjoyable. Seventeen participants preferred the *interactive fly-through* the most. They liked being able to look in the direction they were interested in and control flight speed using the controller. Two participants preferred *passive fly-through* the most. The reason was that the VR did not suit their taste, and they felt a little bit disorientated after the task in VR, but they liked the possibility of seeing the pictures aligned in the virtual terrain model. Two respondents preferred *GPS slideshow* the most since they felt it had been the most helpful to fulfill their task. In summary, *interactive fly-through* is the most enjoyable mode of presentation according to our evaluation.

8.3.6 Discussion

We have measured the subjects' ability to estimate camera position and orientation of a previously unseen photograph based on what they learned from the presentation. We further evaluated the subjects' enjoyment of different presentation modes and the sense of presence they confer. The use of four datasets of different difficulty posed a challenge in the evaluation since we needed to normalize positional errors to compare the differences between the presentation modes.

The results suggest that *GPS slideshow* is likely the best mode for the position estimation task. We suspect that this is because the mode of presentation—markers on a map—is so close to the evaluation task that the effect of recall may dominate that of the genuine sense of spatial orientation. The use of the same modality then leads to marginally better results over *passive fly-through* and *interactive fly-through*.

According to our measurements, it seems that the length of the fly-through and terrain complexity affect the learning effect of the *interactive fly-through*. For a short and easy trip, such as in the *Tahoe* dataset, the *interactive fly-through* scored slightly better than *GPS slideshow* in terms of position and heading, but on more complicated datasets, such as *Nepal*, the *GPS slideshow* performed better. This observation suggests that users get confused when watching large, complicated presentations.

In terms of enjoyment, the *interactive fly-through* mode was preferred by 17 out of 21 participants. The main listed reason was the possibility of looking around freely. The *presence* evaluation also suggests that the users feel more immersion in this mode than in the *GPS slideshow*.

Based on these results, we believe that while the *GPS slideshow* is somewhat better for the quantitative tasks, as it can directly display the queried information, the immersive modes convey an experience closer to that of actually doing the hike in the real-world space. We suspect that if we had included a real-world hike as a mode of presentation, the users would face similar issues in the evaluation as they did with the *interactive fly-through*, as the sense of spatial proprioception acquired by the first-hand experience may not necessarily map to an accurate knowledge of the spatial layout. It would be possible to verify this analogy with an experiment where we would have participants view an *immersive fly-through* and then ask them to retrace the same path in real-life without the use of navigation aids, but an experiment such as this would be difficult to perform and ethically problematic.

8.4 Chapter Summary

We present an automatic method for creation of immersive photo fly-through presentations where the images are overlaid on a virtual model of the terrain. We generate these presentations for four datasets from different geographical areas, with both a passive variant based on viewing these images in Google Earth and an interactive variant in a VR viewer.

Further improvements of our pipeline—*e.g.*, optimization of the photo augmentation step by estimating how many and which photographs to download—could be exciting future work. Moreover, projecting the photograph texture onto the terrain during the fly-through is another direction worth exploring.

We compared our immersive presentation modes with two more traditional ones—a slideshow and a slideshow accompanied by a map—in a user study, where we measured user enjoyment, feeling of presence in the outdoor space, and the ability to understand the location and orientation of images in space. We found that in terms of spatial understanding, our modes performed significantly better than a pure *slideshow* and are on par with the *GPS slideshow*, while the VR-based *interactive fly-through* conveyed a superior sense of presence and was preferred as the most enjoyable by the majority of users.

We hope our immersive trip reports can be useful both in private settings, to share the experience of a trip, and in public, where they could be used to share *e.g.*, trip instructions from users familiar with the area to the users who have yet to visit.

Chapter 9

Conclusions

9.1 Thesis Summary

In this thesis, we focused on visual geo-localization in natural environments. We started with a broad literature review and presented a survey of visual geo-localization methods in Chapter 2. Our survey categorized the visual geo-localization methods into three main categories based on the environment—*global*, *city-scale*, and *natural*. According to our survey, visual geo-localization in the *natural* environment was the least studied category. Although *natural* environments include various scene types—plains, deserts, mountains, oceans, forests, taiga and tundra, and many others, previous work focused mainly on mountains. According to our survey, visual geo-localization in mountainous environments was, however, far from being solved. In light of these facts, we narrowed down this thesis's focus to visual geo-localization in mountainous scenes. Specifically, we focus on estimating camera orientation and position to allow precise alignment of photographs with the terrain model.

We also reviewed the most common datasets and evaluation practices relevant to visual geo-localization in Chapter 3. From this literature review, we concluded that although there are many datasets for visual localization in urbanized areas, datasets with images precisely aligned to the terrain model in mountainous environments are sparse. The only publicly available dataset of this kind was the Venturi Mountain Dataset [142].

9.1.1 Datasets

We created a novel dataset, GeoPose3K, which we presented in Chapter 4. Since all photographs in this dataset are precisely aligned with the terrain model and were manually verified, this dataset was an essential resource for training, evaluating, and comparing our novel methods. To foster the future research of visual geo-localization in natural areas, we also provided a baseline evaluation of a horizon line-based localization method by Saurer *et al*. [163], denoted in this thesis as HLoc.

However, a single dataset was not enough to convey the greedy nature of deep learning approaches. To build large-scale image-based datasets precisely aligned with the terrain model, we experimented with two SfM-based approaches. The reconstruct-then-align approach first reconstructs the 3D scene which is subsequently aligned with the terrain model (Sec. 5.2). Although this approach reconstructed some areas frequently visited by tourists, such as the Matterhorn in the European Alps and Yosemite Valley in the USA, it failed to reconstruct many, not so often photographed scenes. Usually, we could not fully reconstruct a scene due to drift, which was difficult to avoid in complex outdoor scenes with unknown camera intrinsics, varying camera models, frequent occlusions, seasonal, weather, and illumination changes.

To address this issue, we developed a novel method presented in Sec. 5.3, SfM with terrain reference, which uses synthetically rendered ground-level images with known camera parameters to restrict the reconstruction. To our knowledge this method is the first to combine SfM reconstruction with rendered images and was a crucial step to get enough training images to train our camera pose estimation method. In general, when the density and quality of the photographs is high (at least 10 photographs which see the same scene from different viewpoints per square kilometer), it might be easier to use the two-step reconstruct-then-align approach. With lower densities it is unlikely the scene will be reconstructed properly, and our novel SfM with terrain reference shall be used.

9.1.2 Camera orientation estimation

In Chapter 6, we introduced an improved camera orientation estimation framework called Confidence Fusion (CF). Confidence Fusion is based on a spherical cross-correlation of individual inputs that are fused to obtain a single estimate. Confidence Fusion takes features (like edges or semantic segmentation) extracted from an input photograph and a synthetically rendered spherical panorama. We used Confidence Fusion to illustrate that using semantic segmentation of natural features—like forests, glaciers, water bodies, and sky—for cross-domain outdoor scenes is complementary to using edges. We also compared our CF method with the horizon line-based approach HLoc by Saurer *et al.* [163]. We found out, that using our CF method which leverages jointly the edge and semantic segmentation features is much more robust compared to matching only the horizon line. The robustness of our CF method resides mainly in combination of high-frequency edges, and low-frequency semantic segmentation information.

It is advantageous, that our CF method does not require the satellite orthophoto imagery. It requires only the DEM and the semantic segmentation map which is publicly available worldwide, from the open source OpenStreetMap project. On the other hand, the precision and completness of the semantic segmentation map may vary across different parts of the world. According to our experience, the data are fairly complete in the European Alps, but in the US territory some parts of the map are missing, *e.g.*, forested areas are too sparse in Yosemite Valley, USA.

9.1.3 Camera pose estimation

In Chapter 7 we introduced our approach to camera pose estimation with the use of cross-domain keypoint descriptors. In this work, we covered the DEM with a satellite orthophoto texture. Our goal was to find a full camera pose for a single image relative to the terrain using Perspective-*n*-Point (PnP). We trained a cross-domain descriptor based on two branch CNN with a shared trunk to address the cross-domain feature matching problem using the datasets we built in Chapters 4 and 5. Our experiments illustrate that training using our cross-domain data is important for achieving state-of-the-art performance on the task of camera pose estimation using the terrain model. Furthermore, our tiny and mobile friendly architecture performs similarly to much deeper state-of-the-art CNNs, despite that both were trained with the same procedure and data.

Our camera pose estimation method also performs favorably on the task of camera orientation estimation. We compared it to our Confidence Fusion method, which estimates camera orientation by comparing the photograph's edges and semantic segmentation with synthetically rendered silhouettes and semantic segments from the terrain model. According to our experiments, our keypoint-based method estimates the same or better camera orientation than our CF method.

A slight disadvantage of our keypoint-based approach is that it needs the satellite orthophoto texture, which contains essential cues for comparing local regions between the photograph and the terrain. Although the satellite orthophoto textures are available for the whole Earth's surface, they might be costly in the resolution of 5 m/px with which we obtained the best results. According to our experience, the cross-domain descriptor could also be trained to use only the depth map, normal map and silhouettes. However, in this case the number of successfully matched keypoints is significantly lower, since the descriptive regions are located only in the neighborhood of edges and depth discontinuities, which cover the image sparsely unlike the satellite texture.

9.1.4 Photography presentation

In Chapter 8 we proposed an approach to generate presentations of photographs aligned with a terrain model. We proposed two modes of presentation, a passive mode for computer screens, and an interactive mode for virtual reality presentations. Using our method, users may easily create virtual hikes in the form of a fly-through and showcase their photographs in the context of the virtual terrain to others. Our user study indicates that our approach is more enjoyable compared to classical slideshows. Our analysis also shows that our presentation method helps users to self-localize and better understand yet unvisited scenes.

Originally, our method for automatic fly-through generation used the SfM to first reconstruct a sparse point cloud of the scene which was subsequently aligned with the terrain model. Using this process, we were able to align a reasonable number of user's photographs with the terrain under the assumption that the scene was densely covered with the photographs and contained enough distinctive landmarks. An example of such a scene might be the Yosemite Valley in the USA. However, such SfM reconstruction is expensive and prone to drift, especially in the mountainous areas. Both our camera orientation estimation (Sec. 6), and our keypoint-based camera pose estimation (Sec. 7) are drop-in replacements for precise photo-to-terrain alignment which avoid the computationally expensive SfM reconstruction.

9.2 Suggestions for Future Work

In this thesis we contributed towards a reproducible research of visual geo-localization based on the comparison and matching of the photograph with the terrain model. We collected large datasets of automatically aligned photographs with the terrain model, which are suitable for training novel machine and deep learning models. With a lot of manual effort, we carefully created the GeoPose3K dataset suitable for method evaluation. We contributed in-depth evaluations of existing method for horizon-based localization (HLoc) by Saurer *et al*. [163]. We proposed a novel Confidence Fusion (CF) method for camera orientation estimation based on fusing confidences estimated by matching semantic segmentations and edges. To estimate position and orientation of the camera, we devised a novel cross-domain descriptor powered by a compact CNN architecture. All our proposed methods were carefully evaluated using the GeoPose3K dataset, which allowed us to compare our approaches. We also illustrated, that our methods can be applied to create novel immersive experiences by browsing photographs in virtual reality.

We believe that these contributions will stimulate a reproducible research of visual geo-localization in outdoor, natural environments in the future. However, some problems in visual geo-localization based on terrain models remain unsolved and we keep them as a future work. Let us briefly introduce few examples.

Depth and normals estimation. Our datasets we created throughout this thesis, the GeoPose3K and the SfM-based datasets, contain pixel-level annotations of the absolute depth and direction of surface normals. Ahmad *et al.* [4] already used the GeoPose3K dataset for training horizon line detection, and we used it in Chapter 6 for training semantic segmentation. Therefore we anticipate that these datasets are readily usable for training depth and surface normal predictors. Once trained to estimate a metric depth or surface normals similar to the terrain model, we could subsequently use such estimators for photography enhancement in computational photography applications [99, 30], or visual geo-localization.

Large scale localization. This thesis proposed a novel method for 3-DOF camera orientation estimation with a known camera position (Chapter 6) and a novel cross-domain descriptor allowing full 6-DOF camera pose estimation using the Perspective-*n*-Point (PnP) method (Chapter 7). However, large scale position estimation in outdoor environments still awaits in-depth research. According to our literature review in Chapter 2, outdoor visual geo-localization is problematic due to sparse coverage by photographs. The photographs are clustered at locations frequently visited by tourists; however, large not so popular areas are covered by photographs only sporadically. The sparse coverage is why we need to find other solutions to create visual databases. Researchers solved this problem by rendering a terrain from a DEM and using horizon-line features to create geospatial visual databases [190, 189, 186, 214, 200, 37, 163]. According to the results of HLoc, our implementation of Saurer et al. [163] in Chapter 4, localization based on the horizon line feature is dependent on the horizon line detection precision. However, a precise and fully automatic detection of the horizon line is a challenge on its own [5, 4, 144].

On the other hand, in Chapter 6, we illustrated that semantic segmentation helps camera orientation estimation and adds information complementary to edge-based features. A promising future research direction would be to incorporate the semantic segmentation to a large scale visual localization approach to complement the horizon line features. The DEM texture with a satellite orthophoto map could be, in theory, used for large scale visual localization as well. Either training a global descriptor for image retrieval or utilizing our existing local descriptor presented in Chapter 7 with quantization-based methods such as Bag of Words (BOW) [180], Vector of Locally Agreggated Descriptors (VLAD) [86], or Aggregated Selective Match Kernel (ASMK) [193] could be an interesting future research direction.

Crowd-sourced enhancement of terrain textures. In Chapters 5, 6, and 7, we proposed algorithms for the precise alignment of photographs with a terrain model. We believe that an interesting future application could improve the terrain texture using the ground-level photos aligned with the terrain. We could collect photographs from many users, align, and re-project them on the terrain surface to improve the terrain visualization quality.

Such an approach could significantly help overcome low resolution of orthophoto imagery at locations with a steep terrain slope. However, developing such an application brings further research challenges, such as choosing images with a similar season, weather, and illumination so that they could be color mapped and blended into a single texture.

Bibliography

[1] AGARWAL, S., SNAVELY, N., SIMON, I., SEITZ, S. M. and SZELISKI, R. Building Rome in a Day. In: *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*. New York, NY, USA: IEEE, 2009, p. 72–79. DOI: 10.1109/ICCV.2009.5459148. ISBN 978-1-4244-4420-5.

[2] AGUILERA, C. A., AGUILERA, F. J., SAPPA, A. D. and TOLEDO, R. Learning Cross-Spectral Similarity Measures with Deep Convolutional Neural Networks. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. New York, NY, USA: IEEE, 2016, p. 267–275. DOI: 10.1109/CVPRW.2016.40. ISBN 978-1-5090-1438-5.

[3] AGUILERA, C. A., SAPPA, A. D., AGUILERA, C. and TOLEDO, R. Cross-Spectral Local Descriptors via Quadruplet Network. *Sensors*. MDPI. 2017, vol. 17, no. 4, p. 1–14. DOI: 10.3390/s17040873. ISSN 14248220.

[4] AHMAD, T., CAMPR, P., ČADIK, M. and BEBIS, G. Comparison of Semantic Segmentation Approaches for Horizon/Sky Line Detection. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. New York, NY, USA: IEEE, 2017, p. 4436–4443. DOI: 10.1109/IJCNN.2017.7966418. ISBN 978-1-5090-6183-9.

[5] AHMAD, T., BEBIS, G., NICOLESCU, M., NEFIAN, A. and FONG, T. An Edge-Less Approach to Horizon Line Detection. In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. New York, NY, USA: IEEE, 2015, p. 1095–1102. DOI: 10.1109/ICMLA.2015.67. ISBN 978-1-5090-0287-0.

[6] AJGL, J. and SIMANDL, M. Design of a Robust Fusion of Probability Densities. In: *Proceedings of the American Control Conference*. New York, NY, USA: IEEE, 2015, 2015-July, p. 4204–4209. DOI: 10.1109/ACC.2015.7171989. ISBN 978-1-4799-8684-2.

[7] ARANDJELOVIĆ, R., GRONAT, P., TORII, A., PAJDLA, T. and SIVIC, J. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. New York, NY, USA: IEEE, 2016, p. 5297–5307. DOI: 10.1109/CVPR.2016.572. ISBN 978-1-4673-8851-1.

[8] ARDESHIR, S., ZAMIR, A. R., TORROELLA, A. and SHAH, M. GIS-Assisted Object Detection and Geospatial Localization. In: FLEET, D., PAJDLA, T., SCHIELE, B. and TUYTELAARS, T., ed. *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI.* Cham, Switzerland: Springer International Publishing, 2014, vol. 8694, p. 602–617. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-10599-4_39. ISBN 978-3-319-10598-7.

[9] ARMAGAN, A., HIRZER, M., ROTH, P. M. and LEPETIT, V. Learning to Align Semantic Segmentation and 2.5D Maps for Geolocalization. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* New York, NY, USA: IEEE, July 2017, p. 4590–4597. DOI: 10.1109/CVPR.2017.488. ISSN 978-1-5386-0458-8.

[10] AUBRY, M., RUSSELL, B. C. and SIVIC, J. Painting-to-3D Model Alignment via Discriminative Visual Elements. *ACM Trans. Graph.* New York, NY, USA: ACM. 2014, vol. 33, no. 2, p. 14:1–14:14. DOI: 10.1145/2591009. ISSN 0730-0301.

[11] AVRITHIS, Y., KALANTIDIS, Y., TOLIAS, G. and SPYROU, E. Retrieving Landmark and Non-landmark Images from Community Photo Collections. In: *Proceedings of the 18th ACM International Conference on Multimedia.* New York, NY, USA: ACM, 2010, p. 153–162. MM '10. DOI: 10.1145/1873951.1873973. ISBN 978-1-60558-933-6.

[12] BAATZ, G., SAURER, O., KÖSER, K. and POLLEFEYS, M. Leveraging Topographic Maps for Image to Terrain Alignment. In: *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on.* New York, NY, USA: IEEE, October 2012, p. 487–492. DOI: 10.1109/3DIMPVT.2012.33. ISBN 978-1-4673-4470-8.

[13] BAATZ, G., KÖSER, K., CHEN, D., GRZESZCZUK, R. and POLLEFEYS, M. Handling Urban Location Recognition as a 2D Homothetic Problem. In: DANIILIDIS, K., MARAGOS, P. and PARAGIOS, N., ed. *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 6316, p. 266–279. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-15567-3_20. ISBN 978-3-642-15567-3.

[14] BAATZ, G., SAURER, O., KÖSER, K. and POLLEFEYS, M. Large Scale Visual Geo-Localization of Images in Mountainous Terrain. In: FITZGIBBON, A., LAZEBNIK, S., PERONA, P., SATO, Y. and SCHMID, C., ed. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7573, p. 517–530.

Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-33709-3_37. ISBN 978-3-642-33709-3.

[15] BABOUD, L., ČADÍK, M., EISEMANN, E. and SEIDEL, H.-P. Automatic Photo-to-terrain Alignment for the Annotation of Mountain Pictures. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2011, p. 41–48. DOI: 10.1109/CVPR.2011.5995727. ISBN 978-1-4577-0394-2.

[16] BALABANOVIĆ, M., CHU, L. L. and WOLFF, G. J. Storytelling with Digital Photographs. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2000, p. 564—–571. CHI '00. DOI: 10.1145/332040.332505. ISBN 1581132166.

[17] BALNTAS, V., LENC, K., VEDALDI, A., TUYTELAARS, T., MATAS, J. et al. $\mathbb{H}$-Patches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. New York, NY, USA: [b.n.]. 2020, vol. 42, no. 11, p. 2825–2841. DOI: 10.1109/TPAMI.2019.2915233. ISSN 0162-8828.

[18] BANSAL, M., SAWHNEY, H. S., CHENG, H. and DANIILIDIS, K. Geo-localization of Street Views with Aerial Image Databases. In: *Proceedings of the 19th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2011, p. 1125–1128. MM '11. DOI: 10.1145/2072298.2071954. ISBN 978-1-4503-0616-4.

[19] BARUCH, E. B. and KELLER, Y. *Multimodal matching using a Hybrid Convolutional Neural Network* [online]. ArXiv, December 2019. [cit. 2020-12-02]. Available at: https://arxiv.org/abs/1810.12941.

[20] BAY, H., TUYTELAARS, T. and VAN GOOL, L. SURF: Speeded Up Robust Features. In: LEONARDIS, A., BISCHOF, H. and PINZ, A., ed. *Computer Vision – ECCV 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 3951, p. 404–417. Lecture Notes in Computer Science. DOI: 10.1007/11744023_32. ISBN 978-3-540-33833-8.

[21] BEDERSON, B. B. and BOLTMAN, A. Does Animation Help Users Build Mental Maps of Spatial Information? In: *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99)*. New York, NY, USA: IEEE, October 1999, p. 28–35. DOI: 10.1109/INFVIS.1999.801854. ISBN 0-7695-0431-0.

[22] BEHRINGER, R. Improving Registration Precision Through Visual Horizon Silhouette Matching. In: *Proceedings of the International Workshop on Augmented Reality:*

Placing Artificial Objects in Real Scenes. Natick, MA, USA: A. K. Peters, Ltd., 1999, p. 225–232. IWAR '98. ISBN 1-56881-098-9.

[23] BERGAMO, A., SINHA, S. N. and TORRESANI, L. Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2013, p. 763–770. DOI: 10.1109/CVPR.2013.104. ISBN 978-0-7695-4989-7.

[24] BOWRING, B. R. Transformation From Spatial to Geographical Coordinates. *Survey Review*. Taylor & Francis. 1976, vol. 23, no. 181, p. 323–327.

[25] BRACHMANN, E. and ROTHER, C. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. New York, NY, USA: IEEE, 2018, p. 4654–4662. DOI: 10.1109/CVPR.2018.00489. ISBN 978-1-5386-6421-6.

[26] BRACHMANN, E., KRULL, A., NOWOZIN, S., SHOTTON, J., MICHEL, F. et al. DSAC - Differentiable RANSAC for camera localization. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. New York, NY, USA: IEEE, 2017, p. 2492–2500. DOI: 10.1109/CVPR.2017.267. ISBN 978-1-5386-0458-8.

[27] BROWN, M. and LOWE, D. G. Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets. In: *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*. New York, NY, USA: IEEE, 2005, p. 56–63. DOI: 10.1109/3DIM.2005.81. ISBN 0769523277.

[28] BRUBAKER, M. A., GEIGER, A. and URTASUN, R. Lost! Leveraging the Crowd for Probabilistic Visual Self-Localization. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2013, p. 3057–3064. DOI: 10.1109/CVPR.2013.393. ISBN 1063-6919.

[29] BUJNAK, M., KUKELOVA, Z. and PAJDLA, T. A general solution to the P4P problem for camera with unknown focal length. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. New York, NY, USA: IEEE, 2008, p. 1–8. DOI: 10.1109/CVPR.2008.4587793. ISBN 978-1-4244-2242-5.

[30] ČADÍK, M., SÝKORA, D. and LEE, S. Automated Outdoor Depth-Map Generation and Alignment. *Computers & Graphics*. Amsterdam, Netherlands: Elsevier B.V. 2018, vol. 74, p. 109–118. DOI: 10.1016/j.cag.2018.05.001. ISSN 0097-8493.

[31] Čadík, M., Vašíček, J., Hradiš, M., Radenović, F. and Chum, O. Camera Elevation Estimation from a Single Mountain Landscape Photograph. In: Xianghua Xie, M. W. J. and Tam, G. K. L., ed. *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, September 2015, p. 30.1–30.12. DOI: 10.5244/C.29.30. ISBN 1-901725-53-7.

[32] Castaldo, F., Zamir, A., Angst, R., Palmieri, F. and Savarese, S. Semantic Cross-View Matching. In: *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. New York, NY, USA: IEEE, 2015, p. 1044–1052. DOI: 10.1109/ICCVW.2015.137. ISBN 9781467383905.

[33] Chelaramani, S., Muthireddy, V. and Jawahar, C. V. An Interactive Tour Guide for a Heritage Site. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. New York, NY, USA: IEEE, October 2017, p. 2943–2952. DOI: 10.1109/ICCVW.2017.347. ISBN 978-1-5386-1035-0.

[34] Chen, D. M., Baatz, G., Köser, K., Tsai, S. S., Vedantham, R. et al. City-Scale Landmark Identification on Mobile Devices. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2011, p. 737–744. DOI: 10.1109/CVPR.2011.5995610. ISBN 978-1-4577-0394-2.

[35] Chen, J. and Tian, J. Real-time multi-modal rigid registration based on a novel symmetric-SIFT descriptor. *Progress in Natural Science*. Beijing, China: National Natural Science Foundation of China and Chinese Academy of Sciences. 2009, vol. 19, no. 5, p. 643–651. DOI: 10.1016/j.pnsc.2008.06.029. ISSN 10020071.

[36] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. New York, NY, USA: IEEE. 2018, vol. 40, no. 4, p. 834–848. DOI: 10.1109/TPAMI.2017.2699184. ISSN 0162-8828.

[37] Chen, Y., Qian, G., Gunda, K., Gupta, H. and Shafique, K. Camera Geolocation from Mountain Images. In: *2015 18th International Conference on Information Fusion*. New York, NY, USA: IEEE, 2015, p. 1587–1596. ISBN 978-0-9824-4386-6.

[38] Chippendale, P., Zanin, M. and Andreatta, C. Collective Photography. In: *CVMP 2009 - The 6th European Conference for Visual Media Production*. 2009, p. 188–194. DOI: 10.1109/CVMP.2009.30. ISBN 9780769538938.

[39] CHU, H., GALLAGHER, A. and CHEN, T. GPS Refinement and Camera Orientation Estimation from a Single Image and a 2D Map. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. New York, NY, USA: IEEE, June 2014, p. 171–178. DOI: 10.1109/CVPRW.2014.31. ISBN 978-1-4799-4308-1.

[40] CONTE, G. and DOHERTY, P. Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information. *EURASIP Journal on Advances in Signal Processing*. Springer Nature. 2009, vol. 2009, no. 1, p. 387–308. ISSN 1687-6180.

[41] CORDTS, M., OMRAN, M., RAMOS, S., REHFELD, T., ENZWEILER, M. et al. The Cityscapes Dataset for Semantic Urban Scene Understanding. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. New York, NY, USA: IEEE, 2016, p. 3213–3223. DOI: 10.1109/CVPR.2016.350. ISBN 978-1-4673-8852-8.

[42] CRANDALL, D., OWENS, A., SNAVELY, N. and HUTTENLOCHER, D. Discrete-Continuous Optimization for Large-Scale Structure from Motion. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2011, p. 3001–3008. DOI: 10.1109/CVPR.2011.5995626. ISBN 978-1-4577-0394-2.

[43] DALVANDI, A., RIECKE, B. E. and CALVERT, T. Panoramic Video Techniques for Improving Presence in Virtual Environments. In: *Proceedings of the 17th Eurographics Conference on Virtual Environments & Third Joint Virtual Reality*. Aire-la-Ville, Switzerland: Eurographics Association, 2011, p. 103–110. EGVE - JVRC'11. DOI: 10.2312/EGVE/JVRC11/103-110. ISBN 978-3-905674-33-0.

[44] DIECK, M. C. tom and JUNG, T. H. Value of augmented reality at cultural heritage sites: A stakeholder approach. *Journal of Destination Marketing & Management*. Amsterdam, Netherlands: Elsevier B.V. 2017, vol. 6, no. 2, p. 110–117. DOI: 10.1016/j.jdmm.2017.03.002. ISSN 2212-571X. Special edition on Digital Destinations.

[45] DOLLÁR, P. and ZITNICK, C. L. Fast Edge Detection Using Structured Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. New York, NY, USA: IEEE. 2015, vol. 37, no. 8, p. 1558–1570. DOI: 10.1109/TPAMI.2014.2377715. ISSN 0162-8828.

[46] DU, R., BISTA, S. and VARSHNEY, A. Video Fields: Fusing Multiple Surveillance Videos into a Dynamic Virtual Environment. In: *Proceedings of the 21st International Conference on Web3D Technology*. New York, NY, USA: ACM, 2016, p. 165–172. Web3D '16. DOI: 10.1145/2945292.2945299. ISBN 978-1-4503-4428-9.

[47] DU, R. and VARSHNEY, A. Social Street View: Blending Immersive Street Views with Geo-tagged Social Media. In: *Proceedings of the 21st International Conference on Web3D*

Technology. New York, NY, USA: ACM, 2016, p. 77–85. Web3D '16. DOI: 10.1145/2945292.2945294. ISBN 978-1-4503-4428-9.

[48] DUSMANU, M., ROCCO, I., PAJDLA, T., POLLEFEYS, M., SIVIC, J. et al. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New York, NY, USA: IEEE, 2019, p. 8084–8093. DOI: 10.1109/CVPR.2019.00828. ISBN 978-1-7281-3294-5.

[49] EN, S., LECHERVY, A. and JURIE, F. TS-NET: Combining Modality Specific and Common Features for Multimodal Patch Matching. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. New York, NY, USA: IEEE, 2018, p. 3024–3028. DOI: 10.1109/ICIP.2018.8451804. ISBN 978-1-4799-7062-9.

[50] EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K. I., WINN, J. and ZISSERMAN, A. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*. Berlin, Germany: Springer Science+Business Media, LLC. 2010, vol. 88, no. 2, p. 303–338. DOI: 10.1007/s11263-009-0275-4. ISSN 09205691.

[51] FATTAL, R. Dehazing Using Color-Lines. *ACM Trans. Graph.* New York, NY, USA: ACM. December 2014, vol. 34, no. 1, p. 13:1–13:14. DOI: 10.1145/2651362. ISSN 0730-0301.

[52] FEDOROV, R., FRAJBERG, D. and FRATERNALI, P. A Framework for Outdoor Mobile Augmented Reality and Its Application to Mountain Peak Detection. In: DE PAOLIS, L. T. and MONGELLI, A., ed. *Augmented Reality, Virtual Reality, and Computer Graphics*. Cham, Germany: Springer International Publishing, 2016, p. 281–301. DOI: 10.1007/978-3-319-40621-3_21. ISBN 978-3-319-40621-3.

[53] FEDOROV, R., FRATERNALI, P. and TAGLIASACCHI, M. Mountain peak identification in visual content based on coarse Digital Elevation Models. In: *Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data*. New York, NY, USA: ACM, 2014, p. 7–11. DOI: 10.1145/2661821.2661825. ISBN 9781450331234.

[54] FIRMAN, M. RGBD Datasets: Past, Present and Future. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. New York, NY, USA: IEEE, 2016, p. 661–673. DOI: 10.1109/CVPRW.2016.88. ISBN 978-1-5090-1438-5.

[55] FISCHLER, M. and BOLLES, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applicatlons to Image Analysis and Automated Cartography. *Communications of the ACM*. New York, NY, USA: ACM. 1981, vol. 24, no. 6, p. 381–395. DOI: 10.1145/358669.358692. ISSN 0001-0782.

[56] FLATOW, D., NAAMAN, M., XIE, K. E., VOLKOVICH, Y. and KANZA, Y. On the Accuracy of Hyper-local Geotagging of Social Media Content. In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2015, p. 127–136. WSDM '15. DOI: 10.1145/2684822.2685296. ISBN 978-1-4503-3317-7.

[57] GALLAGHER, A., JOSHI, D., YU, J. and LUO, J. Geo-location Inference from Image Content and User Tags. In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, p. 55–62. DOI: 10.1109/CVPR.2009.5204168. ISBN 9781424439911.

[58] GE, Y., WANG, H., ZHU, F., ZHAO, R. and LI, H. Self-supervising Fine-Grained Region Similarities for Large-Scale Image Localization. In: VEDALDI, A., BISCHOF, H., BROX, T. and FRAHM, J.-M., ed. *Computer Vision – ECCV 2020*. Cham, Germany: Springer International Publishing, 2020, vol. 12349, p. 369–386. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-58548-8_22. ISBN 978-3-030-58548-8.

[59] GEIGER, A., LENZ, P., STILLER, C. and URTASUN, R. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*. London, UK: SAGE Publishing. 2013, vol. 32, no. 11, p. 1231–1237. DOI: 10.1177/0278364913491297. ISSN 0278-3649.

[60] GEORGAKIS, G., KARANAM, S., WU, Z., ERNST, J. and KOŠECKÁ, J. End-to-End Learning of Keypoint Detector and Descriptor for Pose Invariant 3D Matching. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. New York, NY, USA: IEEE, 2018, p. 1965–1973. DOI: 10.1109/CVPR.2018.00210. ISBN 978-1-5386-6421-6.

[61] GEORGAKIS, G., KARANAM, S., WU, Z. and KOSECKA, J. Learning Local RGB-to-CAD Correspondences for Object Pose Estimation. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. New York, NY, USA: IEEE, 2019, p. 8966–8975. DOI: 10.1109/ICCV.2019.00906. ISBN 978-1-7281-4804-5.

[62] GESCH, D., OIMOEN, M., GREENLEE, S., NELSON, C., STEUCK, M. et al. The National Elevation Dataset. *Photogrammetric Engineering and Remote Sensing*. Bethesda, Maryland, USA: ASPRS. 2002, vol. 68, p. 5–11. ISSN 0099-1112.

[63] GRAHAM, A., GARCIA MOLINA, H., PAEPCKE, A. and WINOGRAD, T. Time as Essence for Photo Browsing Through Personal Digital Libraries. In: *Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries - JCDL '02*. 2002, p. 326. DOI: 10.1145/544220.544301. ISBN 1581135130.

[64] GRIFFITH, S., CHAHINE, G. and PRADALIER, C. Symphony Lake Dataset. *International Journal of Robotics Research (IJRR)*. London, UK: SAGE Publishing. September 2017, vol. 36, no. 11, p. 1151–1158. DOI: 10.1177/0278364917730606. ISSN 0278-3649.

[65] GRZESZCZUK, R., KOŠECKÁ, J., VEDANTHAM, R. and HILE, H. Creating Compact Architectural Models by Geo-registering Image Collections. In: *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*. New York, NY, USA: IEEE, 2009, p. 1718–1725. DOI: 10.1109/ICCVW.2009.5457490. ISBN 978-1-4244-4442-7.

[66] HAAN, G. de, PIGUILLET, H. and POST, F. Spatial Navigation for Context-Aware Video Surveillance. *IEEE Computer Graphics and Applications*. September 2010, vol. 30, no. 5, p. 20–31. DOI: 10.1109/MCG.2010.64. ISSN 0272-1716.

[67] HAKEEM, A., VEZZANI, R., SHAH, M. and CUCCHIARA, R. Estimating Geospatial Trajectory of a Moving Camera. In: *Proceedings of the 18th International Conference on Pattern Recognition*. New York, NY, USA: IEEE, 2006, vol. 2, p. 82–87. DOI: 10.1109/ICPR.2006.499. ISBN 0-7695-2521-0.

[68] HAKLAY, M. M. and WEBER, P. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*. Piscataway, NJ, USA: IEEE Educational Activities Department. October 2008, vol. 7, no. 4, p. 12–18. DOI: 10.1109/MPRV.2008.80. ISSN 1536-1268.

[69] HAMMOUD, R. I., KUZDEBA, S. A., BERARD, B., TOM, V., IVEY, R. et al. Overhead-based image and video geo-localization framework. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Los Alamitos, CA, USA: IEEE Computer Society, 2013, p. 320–327. DOI: 10.1109/CVPRW.2013.55. ISBN 9780769549903.

[70] HAO, Q., CAI, R., LI, Z., ZHANG, L., PANG, Y. et al. 3D Visual Phrases for Landmark Recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2012, p. 3594–3601. DOI: 10.1109/CVPR.2012.6248104. ISBN 9781467312264.

[71] HARADA, S., NAAMAN, M., SONG, Y. J., WANG, Q. and PAEPCKE, A. Lost in Memories: Interacting with Photo Collections on PDAs. In: *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*. New York, NY, USA: IEEE, 2004, p. 325. DOI: 10.1109/JCDL.2004.1336143. ISBN 1-58113-832-6.

[72] HARTLEY, R. and ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Cambridge, UK: Cambridge University Press, 2004. 655 p. ISBN 9780874216561.

[73] HARWOOD, B., VIJAY KUMAR, B. G., CARNEIRO, G., REID, I. and DRUMMOND, T. Smart Mining for Deep Metric Learning. In: *Proceedings of the IEEE International Conference on Computer Vision*. New York, NY, USA: IEEE, 2017. DOI: 10.1109/ICCV.2017.307. ISBN 978-1-5386-1033-6.

[74] HASAN, M., PICKERING, M. R. and JIA, X. Modified SIFT for Multi-modal Remote Sensing Image Registration. In: *2012 IEEE International Geoscience and Remote Sensing Symposium*. New York, NY, USA: IEEE, July 2012, p. 2348–2351. DOI: 10.1109/IGARSS.2012.6351023. ISBN 978-1-4673-1160-1.

[75] HAUGSTVEDT, A. C. and KROGSTIE, J. Mobile Augmented Reality for Cultural Heritage: A Technology Acceptance Study. In: *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. New York, NY, USA: IEEE, November 2012, p. 247–255. DOI: 10.1109/ISMAR.2012.6402563. ISBN 978-1-4673-4660-3.

[76] HAYDAR, M., ROUSSEL, D., MAÏDI, M., OTMANE, S. and MALLEM, M. Virtual and augmented reality for cultural computing and heritage: a case study of virtual exploration of underwater archaeological sites (preprint). *Virtual Reality*. London, UK: Springer-Verlag. November 2011, vol. 15, no. 4, p. 311–327. DOI: 10.1007/s10055-010-0176-4. ISSN 1434-9957.

[77] HAYS, J. and EFROS, A. A. IM2GPS: estimating geographic information from a single image. In: *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition*. New York, NY, USA: IEEE, 2008. ISBN 9781424422432.

[78] HAYS, J. and EFROS, A. A. Large-Scale Image Geolocalization. In: CHOI, J. and FRIEDLAND, G., ed. *Multimodal Location Estimation of Videos and Images*. Switzerland: Springer International Publishing, 2015, chap. Large-Scale Image Geolocalization, p. 41–62. DOI: 10.1007/978-3-319-09861-6. ISBN 978-3-319-09861-6.

[79] HE, K., SUN, J. and TANG, X. Single Image Haze Removal Using Dark Channel Prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Los Alamitos, CA, USA: IEEE Computer Society. 2011, vol. 33, no. 12, p. 2341–2353. DOI: 10.1109/TPAMI.2010.168. ISSN 01628828.

[80] HEINLY, J., SCHÖNBERGER, J. L., DUNN, E. and FRAHM, J.-M. Reconstructing the World* in Six Days. In: *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*. New York, NY, USA: IEEE, 2015, p. 3287–3295. DOI: 10.1109/CVPR.2015.7298949. ISBN 9781467369640.

[81] HILLIGES, O., BAUR, D. and BUTZ, A. Photohelix: Browsing, Sorting and Sharing Digital Photo Collections. In: *Tabletop 2007 - 2nd Annual IEEE International Workshop*

on Horizontal Interactive Human-Computer Systems. Los Alamitos, CA, USA: IEEE Computer Society, 2007, p. 87–94. DOI: 10.1109/TABLETOP.2007.20. ISBN 978-0-7695-2013-1.

[82] IRANI, M. and ANANDAN, P. Robust multi-sensor image alignment. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. New York, NY, USA: IEEE, January 1998, p. 959–966. DOI: 10.1109/ICCV.1998.710832. ISBN 81-7319-221-9.

[83] IRSCHARA, A., ZACH, C., FRAHM, J. M. and BISCHOF, H. From Structure-from-Motion Point Clouds to Fast Location Recognition. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, p. 2599–2606. DOI: 10.1109/CVPRW.2009.5206587. ISBN 978-1-4244-3992-8.

[84] IZBICKI, M., PAPALEXAKIS, E. E. and TSOTRAS, V. J. Exploiting the Earth's Spherical Geometry to Geolocate Images. In: BREFELD, U., FROMONT, E., HOTHO, A., KNOBBE, A., MAATHUIS, M. et al., ed. *Machine Learning and Knowledge Discovery in Databases*. Cham, Germany: Springer International Publishing, 2020, p. 3–19. DOI: 10.1007/978-3-030-46147-8_1. ISBN 978-3-030-46147-8.

[85] JACOBS, N., SATKIN, S., ROMAN, N., SPEYER, R. and PLESS, R. Geolocating Static Cameras. In: *Proceedings of the IEEE International Conference on Computer Vision*. New York, NY, USA: IEEE, 2007, p. 1–6. DOI: 10.1109/ICCV.2007.4408995. ISBN 978-1-4244-1630-1.

[86] JÉGOU, H., DOUZE, M., SCHMID, C. and PÉREZ, P. Aggregating local descriptors into a compact image representation. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2010, p. 3304–3311. DOI: 10.1109/CVPR.2010.5540039. ISBN 9781424469840.

[87] JOHNS, E. and YANG, G. Z. From Images to Scenes: Compressing an Image Cluster into a Single Scene Model for Place Recognition. In: *Proceedings of the IEEE International Conference on Computer Vision*. New York, NY, USA: IEEE, 2011, p. 874–881. DOI: 10.1109/ICCV.2011.6126328. ISBN 978-1-4577-1101-5.

[88] KABSCH, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*. Chester, UK: IUCr. 1976, vol. 32, no. 5, p. 922–923. DOI: 10.1107/S0567739476001873. ISSN 16005724.

[89] KALOGERAKIS, E., VESSELOVA, O., HAYS, J., EFROS, A. A. and HERTZMANN, A. Image Sequence Geolocation with Human Travel Priors. In: *Proceedings of the IEEE*

International Conference on Computer Vision. New York, NY, USA: IEEE, 2009, p. 253–260. DOI: 10.1109/ICCV.2009.5459259. ISBN 978-1-4244-4420-5.

[90] KANEVA, B., SIVIC, J., TORRALBA, A., AVIDAN, S. and FREEMAN, W. T. Infinite Images: Creating and Exploring a Large Photorealistic Virtual Space. *Proceedings of the IEEE*. New York, NY, USA: IEEE. 2010, vol. 98, no. 8, p. 1391–1407. DOI: 10.1109/JPROC.2009.2031133. ISSN 0018-9219.

[91] KANG, H. and SHNEIDERMAN, B. Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder. In: *Proceedings of the 2000 IEEE International Conference on Multimedia and Expo (ICME2000)*. 2000, vol. 3, p. 1539–1542. DOI: 10.1109/ICME.2000.871061. ISBN 0-7803-6536-4.

[92] KELLER, Y. and AVERBUCH, A. Multisensor Image Registration via Implicit Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Los Alamitos, CA, USA: IEEE Computer Society. May 2006, vol. 28, no. 5, p. 794–801. DOI: 10.1109/TPAMI.2006.100. ISSN 1939-3539.

[93] KELM, P., SCHMIEDEKE, S. and SIKORA, T. A Hierarchical, Multi-Modal Approach for Placing Videos on the Map Using Millions of Flickr Photographs. In: *Proceedings of the 2011 ACM Workshop on Social and Behavioural Networked Media Access*. New York, NY, USA: ACM, 2011, p. 15–20. SBNMA '11. DOI: 10.1145/2072627.2072634. ISBN 9781450309905.

[94] KELM, P., SCHMIEDEKE, S. and SIKORA, T. Multi-Modal, Multi-Resource Methods for Placing Flickr Videos on the Map. In:. New York, NY, USA: ACM, 2011, p. 1–8. ICMR '11. DOI: 10.1145/1991996.1992048. ISBN 978-1-4503-0336-1.

[95] KENDALL, A., GRIMES, M. and CIPOLLA, R. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision*. New York, NY, USA: IEEE, 2015, p. 2938–2946. DOI: 10.1109/ICCV.2015.336. ISBN 978-1-4673-8391-2.

[96] KIM, G., MOON, S. and SIGAL, L. Joint Photo Stream and Blog Post Summarization and Exploration. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. New York, NY, USA: IEEE, June 2015, p. 3081–3089. DOI: 10.1109/CVPR.2015.7298927. ISBN 978-1-4673-6964-0.

[97] KLEIN, G. and MURRAY, D. Parallel Tracking and Mapping for Small AR Workspaces. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*. New York, NY, USA: IEEE, 2007. DOI: 10.1109/ISMAR.2007.4538852. ISBN 978-1-4244-1749-0.

[98] KOPF, J., COHEN, M. F. and SZELISKI, R. First-person Hyper-lapse Videos. *ACM Trans. Graph.* New York, NY, USA: ACM. July 2014, vol. 33, no. 4, p. 78:1–78:10. DOI: 10.1145/2601097.2601195. ISSN 0730-0301.

[99] KOPF, J., NEUBERT, B., CHEN, B., COHEN, M., COHEN OR, D. et al. Deep Photo: Model-Based Photograph Enhancement and Viewing. *ACM Trans. Graph.* New York, NY, USA: ACM. 2008, vol. 27, no. 5, p. 1–10. DOI: 10.1145/1409060.1409069. ISSN 0730-0301.

[100] KOŠECKÁ, J. and ZHANG, W. Video Compass. In: HEYDEN, A., SPARR, G., NIELSEN, M. and JOHANSEN, P., ed. *Computer Vision – ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part IV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, vol. 2353, p. 476–490. Lecture Notes in Computer Science. DOI: 10.1007/3-540-47979-1_32. ISBN 978-3-540-47979-6.

[101] KOSTELEC, P. J. and ROCKMORE, D. N. FFTs on the rotation group. *Journal of Fourier Analysis and Applications*. Springer Nature. 2008, vol. 14, no. 2, p. 145–179. DOI: 10.1007/s00041-008-9013-5. ISSN 10695869.

[102] KRALJIC, N. Interactive Video Virtual Tours. In: *Proceedings of the Central European Seminar on Computer Graphics (CESCG)* [online]. April 2008. ISBN 978-3-9502533-0-6. [cit. 2020-12-04]. Available at: https://old.cescg.org/CESCG-2008/papers/Sarajevo-Nermina-Kraljic.pdf.

[103] KUCHELMEISTER, V. and BENNET, J. The Amnesia Atlas. An immersive SenseCam Interface as memory-prosthesis. In: *Proceedings of the 2014 International Conference on Virtual Systems and Multimedia, VSMM 2014*. New York, NY, USA: IEEE, 2014, p. 217–222. DOI: 10.1109/VSMM.2014.7136663. ISBN 978-1-4799-7227-2.

[104] KUSHAL, A., SELF, B., FURUKAWA, Y., GALLUP, D., HERNANDEZ, C. et al. Photo Tours. In: *Proceedings - 2nd Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2012*. New York, NY, USA: IEEE, 2012, p. 57–64. DOI: 10.1109/3DIMPVT.2012.62. ISBN 978-1-4673-4470-8.

[105] KWON, Y. P., KIM, H., KONJEVOD, G. and MCMAINS, S. Dude (Duality descriptor): A robust descriptor for disparate images using line segment duality. In: *2016 IEEE International Conference on Image Processing (ICIP)*. New York, NY, USA: IEEE, September 2016, p. 310–314. DOI: 10.1109/ICIP.2016.7532369. ISBN 978-1-4673-9962-3.

[106] LADICKY, L., RUSSELL, C., KOHLI, P. and TORR, P. H. S. Graph Cut Based Inference with Co-occurrence Statistics. In: *Proceedings of the 11th European Conference on Computer Vision: Part V*. Berlin, Heidelberg: Springer-Verlag, 2010, p. 239–253. ECCV'10. DOI: 10.1007/978-3-642-15555-0_18. ISBN 3-642-15554-5.

[107] LALONDE, J.-F., NARASIMHAN, S. G. and EFROS, A. A. What do the sun and the sky tell us about the camera? *International Journal on Computer Vision*. 2010, vol. 88, no. 1, p. 24–51. DOI: 10.1007/s11263-009-0291-4. ISSN 0920-5691.

[108] LARNAOUT, D., BOURGEOIS, S., GAY BELLILE, V. and DHOME, M. Towards Bundle Adjustment with GIS Constraints for Online Geo-Localization of a Vehicle in Urban Center. In: *Proceedings - 2nd Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2012*. New York, NY, USA: IEEE, 2012, p. 348–355. DOI: 10.1109/3DIMPVT.2012.38. ISBN 978-1-4673-4470-8.

[109] LARNAOUT, D., GAY BELLILE, V., BOURGEOIS, S. and DHOME, M. Vehicle 6-DoF localization based on SLAM constrained by GPS and digital elevation model information. In: *Proceedings of the 2013 20th IEEE International Conference on Image Processing (ICIP)*. New York, NY, USA: IEEE, 2013, p. 2504–2508. DOI: 10.1109/ICIP.2013.6738516. ISBN 978-1-4799-2341-0.

[110] LEPETIT, V., MORENO NOGUER, F. and FUA, P. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*. Berlin, Germany: Springer Science+Business Media, LLC. 2009, vol. 81, no. 2. DOI: 10.1007/s11263-008-0152-6. ISSN 0920-5691.

[111] LEVINSON, J. and THRUN, S. Robust Vehicle Localization in Urban Environments Using Probabilistic Maps. In: *2010 IEEE International Conference on Robotics and Automation*. New York, NY, USA: IEEE, 2010, p. 4372–4378. DOI: 10.1109/ROBOT.2010.5509700. ISBN 978-1-4244-5038-1.

[112] LI, Y., CRANDALL, D. J. and HUTTENLOCHER, D. P. Landmark classification in large-scale image collections. *Proceedings of the IEEE International Conference on Computer Vision*. 2009, p. 1957–1964. ISSN 1550-5499.

[113] LI, Y., SNAVELY, N., HUTTENLOCHER, D. and FUA, P. Worldwide Pose Estimation Using 3D Point Clouds. In: FITZGIBBON, A., LAZEBNIK, S., PERONA, P., SATO, Y. and SCHMID, C., ed. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7572, p. 15–29. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-33718-5_2. ISBN 978-3-642-33718-5.

[114] LI, Y., SNAVELY, N. and HUTTENLOCHER, D. P. Location Recognition Using Prioritized Feature Matching. In: DANIILIDIS, K., MARAGOS, P. and PARAGIOS, N., ed. *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 6312, p. 791–804. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-15552-9_57. ISBN 978-3-642-15552-9.

[115] LI, Z. and SNAVELY, N. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2018, p. 2041–2050. DOI: 10.1109/CVPR.2018.00218. ISBN 978-1-5386-6421-6.

[116] LIN, T. Y., BELONGIE, S. and HAYS, J. Cross-view image geolocalization. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2013, p. 891–898. DOI: 10.1109/CVPR.2013.120. ISBN 978-0-7695-4989-7.

[117] LIN, T.-Y., BELONGIE, S. and HAYS, J. Learning deep representations for ground-to-aerial geolocalization. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition*. New York, NY, USA: IEEE, 2015, p. 5007–5015. DOI: 10.1109/CVPR.2015.7299135. ISBN 978-1-4673-6964-0.

[118] LIN, T. Y., MAIRE, M., BELONGIE, S., HAYS, J., PERONA, P. et al. Microsoft COCO: Common Objects in Context. In: FLEET, D., PAJDLA, T., SCHIELE, B. and TUYTELAARS, T., ed. *Computer Vision – ECCV 2014*. Cham, Germany: Springer International Publishing, 2014, vol. 8693, p. 740–755. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-10602-1_48. ISBN 978-3-319-10601-4.

[119] LIU, L., LI, H. and DAI, Y. Stochastic attraction-repulsion embedding for large scale image localization. In: *Proceedings of the IEEE International Conference on Computer Vision*. Los Alamitos, CA, USA: IEEE Computer Society, 2019, p. 2570–2579. DOI: 10.1109/ICCV.2019.00266. ISBN 978-1-7281-4804-5.

[120] LONG, J., SHELHAMER, E. and DARRELL, T. Fully convolutional networks for semantic segmentation. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2015, p. 3431–3440. DOI: 10.1109/CVPR.2015.7298965. ISBN 978-1-4673-6964-0.

[121] LOWE, D. G. Object Recognition from Local Scale-Invariant Features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Los

Alamitos, CA, USA: IEEE Computer Society, 1999, vol. 2, p. 1150–1157. DOI: 10.1109/ICCV.1999.790410. ISBN 0-7695-0164-8.

[122] LOWE, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. Berlin, Germany: Springer Science+Business Media, LLC. 2004, vol. 60, no. 2, p. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94. ISSN 1573-1405.

[123] MADDERN, W., PASCOE, G., LINEGAR, C. and NEWMAN, P. 1 Year, 1000 km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*. London, UK: SAGE Publishing. 2017, vol. 36, no. 1, p. 3–15. DOI: 10.1177/0278364916679498. ISSN 0278-3649.

[124] MCCURDY, N. J., GRISWOLD, W. G. and LENERT, L. A. RealityFlythrough: enhancing situational awareness for medical response to disasters using ubiquitous video. In:. Bethesda, Maryland, USA: AMIA, 2005, p. 510–514.

[125] MIDDELBERG, S., SATTLER, T., UNTZELMANN, O. and KOBBELT, L. Scalable 6-DOF Localization on Mobile Devices. In: FLEET, D., PAJDLA, T., SCHIELE, B. and TUYTELAARS, T., ed. *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II*. Cham, Germany: Springer International Publishing, 2014, vol. 8690, p. 268–283. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-10605-2_18. ISBN 978-3-319-10605-2.

[126] MISHCHUK, A., MISHKIN, D., RADENOVIĆ, F. and MATAS, J. Working hard to know your neighbor's margins: Local descriptor learning loss. In: *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017, vol. 30, p. 4826–4837. NIPS'17. ISBN 9781510860964.

[127] MISHKIN, D., PERDOCH, M. and MATAS, J. Place Recognition with WxBS Retrieval. In: *CVPR 2015 Workshop on Visual Place Recognition in Changing Environments*. 2015.

[128] MONTEMERLO, M., BECKER, J., BHAT, S., DAHLKAMP, H., DOLGOV, D. et al. Junior: The Stanford entry in the Urban Challenge. *Journal of Field Robotics*. Chichester, UK: John Wiley and Sons Ltd. 2008, vol. 25, no. 9, p. 569–597. DOI: 10.1002/rob. ISSN 1556-4967.

[129] MOTTAGHI, R., CHEN, X., LIU, X., CHO, N. G., LEE, S. W. et al. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2014, p. 891–898. DOI: 10.1109/CVPR.2014.119. ISBN 978-1-4799-5118-5.

[130] MOULON, P., MONASSE, P. and MARLET, R. Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion. In: *2013 IEEE International Conference on Computer Vision*. Los Alamitos, CA, USA: IEEE Computer Society, December 2013, p. 3248–3255. DOI: 10.1109/ICCV.2013.403. ISBN 978-1-4799-2840-8.

[131] MÜLLER BUDACK, E., PUSTU IREN, K. and EWERTH, R. Geolocation Estimation of Photos Using a Hierarchical Model and Scene Classification. In: FERRARI, V., HEBERT, M., SMINCHISESCU, C. and WEISS, Y., ed. *Computer Vision – ECCV 2018*. Cham, Germany: Springer International Publishing, 2018, vol. 11216. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-01258-8_35. ISBN 978-3-030-01258-8.

[132] NAGY, B. A New Method of Improving the Azimuth in Mountainous Terrain by Skyline Matching. *PFG—Journal of Photogrammetry, Remote Sensing and Geoinformation Science*. Springer Nature. 2020. DOI: 10.1007/s41064-020-00093-1. ISSN 2512-2819.

[133] NAVAL, P. C. Camera Pose Estimation by Alignment from a Single Mountain Image. *International Symposium on Intelligent Robotic Systems*. 1998, p. 157–163.

[134] NISTÉR, D. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Los Alamitos, CA, USA: IEEE Computer Society. 2004, vol. 26, no. 6, p. 0756–777. DOI: 10.1109/TPAMI.2004.17. ISSN 0162-8828.

[135] OONUKI, S. and OGI, T. VR BBS Using Immersive Virtual Environment. In: *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*. Los Alamitos, CA, USA: IEEE Computer Society, 2008, p. 1006–1011. DOI: 10.1109/WAINA.2008.119. ISBN 978-0-7695-3096-3.

[136] PATTERSON, G. and HAYS, J. SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2012, p. 2751–2758. DOI: 10.1109/CVPR.2012.6247998. ISBN 978-1-4673-1226-4.

[137] PAUSANIAS. *Graecae descriptio*. C. 150.

[138] PISA, R. da and POLO, M. *Livres des Merveilles du Monde*. C. 1300.

[139] PLATT, J. C. AutoAlbum: Clustering Digital Photographs using Probabilistic Model Merging. In: *2000 Proceedings Workshop on Content-Based Access of Image and Video Libraries*. Los Alamitos, CA, USA: IEEE Computer Society, 2000, p. 96–100. DOI: 10.1109/IVL.2000.853847. ISBN 0-7695-0695-X.

[140] PLATT, J. C., CZERWINSKI, M. and FIELD, B. A. PhotoTOC: Automatic Clustering for Browsing Personal Photographs. In: *ICICS-PCM 2003 - Proceedings of the 2003 Joint Conference of the 4th International Conference on Information, Communications and Signal Processing and 4th Pacific-Rim Conference on Multimedia*. New York, NY, USA: IEEE, 2003, vol. 1, p. 6–10. DOI: 10.1109/ICICS.2003.1292402. ISBN 0-7803-8185-8.

[141] POMERLEAU, F., COLAS, F., SIEGWART, R. and MAGNENAT, S. Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots*. Berlin, Germany: Springer Science+Business Media, LLC. February 2013, vol. 34, no. 3, p. 133–148. DOI: 10.1007/s10514-013-9327-2. ISSN 0929-5593.

[142] PORZI, L., BULÒ, S. R., LANZ, O., VALIGI, P. and RICCI, E. An automatic image-to-DEM alignment approach for annotating mountains pictures on a smartphone. *Machine Vision and Applications*. Berlin, Heidelberg: Springer-Verlag. 2016, vol. 28, p. 101—-115. DOI: 10.1007/s00138-016-0808-0. ISSN 1432-1769.

[143] PORZI, L., BULÓ, S. R., VALIGI, P., LANZ, O. and RICCI, E. Learning Contours for Automatic Annotations of Mountains Pictures on a Smartphone. In: *Proceedings of the International Conference on Distributed Smart Cameras*. New York, NY, USA: ACM, 2014, p. 13:1–13:6. ICDSC '14. DOI: 10.1145/2659021.2659046. ISBN 978-1-4503-2925-5.

[144] PORZI, L., ROTA BULÒ, S. and RICCI, E. A Deeply-Supervised Deconvolutional Network for Horizon Line Detection. In: *Proceedings of the 24th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2016, p. 137—-141. MM '16. DOI: 10.1145/2964284.2967198. ISBN 9781450336031.

[145] PRODUIT, T., TUIA, D., GOLAY, F. and STRECHA, C. Pose Estimation of Landscape Images Using DEM and Orthophotos. In: *2012 International Conference on Computer Vision in Remote Sensing*. New York, NY, USA: IEEE, 2012, p. 209–214. DOI: 10.1109/CVRS.2012.6421262. ISBN 978-1-4673-1272-1.

[146] PROSPERO, C., MUKUNOKI, M., MINOH, M. and IKEDA, K. Estimating Camera Position and Orientation from Geographical Map and Mountain Image. In: *38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers*. 1997, p. 9–16.

[147] RADIANO, O., GRABER, Y., MAHLER, M., SIGAL, L. and SHAMIR, A. Story Albums: Creating Fictional Stories From Personal Photograph Sets. *Computer Graphics Forum*. 2017, vol. 37, no. 1, p. 19–31. DOI: 10.1111/cgf.13099. ISSN 1467-8659.

[148] RAGURAM, R., WU, C., FRAHM, J. M. and LAZEBNIK, S. Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs. *International Journal of*

Computer Vision. Berlin, Germany: Springer Science+Business Media, LLC. 2011, vol. 95, no. 3, p. 213–239. DOI: 10.1007/s11263-011-0445-z. ISSN 0920-5691.

[149] RAMALINGAM, S., BOUAZIZ, S., STURM, P. and BRAND, M. Geolocalization using Skylines from Omni-Images. In: *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, p. 23–30. DOI: 10.1109/ICCVW.2009.5457723. ISBN 978-1-4244-4442-7.

[150] RAMALINGAM, S., BOUAZIZ, S., STURM, P. and BRAND, M. SKYLINE2GPS: Localization in Urban Canyons using Omni-Skylines. In: *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems*. New York, NY, USA: IEEE, 2010, p. 3816–3823. DOI: 10.1109/IROS.2010.5649105. ISBN 978-1-4244-6674-0.

[151] RITCHIE, H. and ROSER, M. *Urbanization* [online]. Our World in Data, September 2018, revised November 2019. [cit. 2020-12-05]. Available at: https://ourworldindata.org/urbanization.

[152] ROBERTSONE, D. and CIPOLLA, R. An Image-Based System for Urban Navigation. In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 2004, p. 84.1–84.10. DOI: doi:10.5244/C.18.84. ISBN 1-901725-25-1.

[153] ROCCO, I., CIMPOI, M., ARANDJELOVIĆ, R., TORII, A., PAJDLA, T. et al. Neighbourhood Consensus Networks. In: BENGIO, S., WALLACH, H., LAROCHELLE, H., GRAUMAN, K., CESA BIANCHI, N. et al., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018, vol. 31, p. 1651–1662. ISBN 9781510884472.

[154] RODDEN, K. and WOOD, K. R. How Do People Manage Their Digital Photographs? In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2003, no. 1, p. 409–416. CHI '03. DOI: 10.1145/642611.642682. ISBN 1581136307. Available at: http://portal.acm.org/citation.cfm?doid=642611.642682.

[155] RUBLEE, E., RABAUD, V., KONOLIGE, K. and BRADSKI, G. ORB: An efficient alternative to SIFT or SURF. In: *Proceedings of the IEEE International Conference on Computer Vision*. Los Alamitos, CA, USA: IEEE Computer Society, 2011, p. 2564–2571. DOI: 10.1109/ICCV.2011.6126544. ISBN 978-1-4577-1101-5.

[156] RUZON, M. A. and TOMASI, C. Color Edge Detection with the Compass Operator. In: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition (Cat. No PR00149). Los Alamitos, CA, USA: IEEE Computer Society, 1999, vol. 2, June, p. 160–166. DOI: 10.1109/CVPR.1999.784624. ISBN 0-7695-0149-4.

[157] SATTLER, T., HAVLENA, M., RADENOVI, F., SCHINDLER, K. and POLLEFEYS, M. Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. New York, NY, USA: IEEE, 2015, p. 2102–2110. DOI: 10.1109/ICCV.2015.243. ISBN 978-1-4673-8391-2.

[158] SATTLER, T., LEIBE, B. and KOBBELT, L. Fast Image-Based Localization using Direct 2D-to-3D Matching. In: *Proceedings of the IEEE International Conference on Computer Vision*. New York, NY, USA: IEEE, 2011, p. 667–674. DOI: 10.1109/ICCV.2011.6126302. ISBN 978-1-4577-1101-5.

[159] SATTLER, T., LEIBE, B. and KOBBELT, L. Improving Image-Based Localization by Active Correspondence Search. In: FITZGIBBON, A., LAZEBNIK, S., PERONA, P., SATO, Y. and SCHMID, C., ed. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7572, p. 752–765. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-33718-5_54. ISBN 978-3-642-33718-5.

[160] SATTLER, T., MADDERN, W., TOFT, C., TORII, A., HAMMARSTRAND, L. et al. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2018, p. 8601–8610. DOI: 10.1109/CVPR.2018.00897. ISBN 978-1-5386-6421-6.

[161] SATTLER, T., WEYAND, T., LEIBE, B. and KOBBELT, L. Image Retrieval for Image-Based Localization Revisited. In: *Procedings of the British Machine Vision Conference 2012*. BMVA Press, 2012, p. 76.1–76.12. DOI: 10.5244/C.26.76. ISBN 1-901725-46-4.

[162] SATTLER, T., ZHOU, Q., POLLEFEYS, M. and LEAL TAIXE, L. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2019, p. 3297–3307. DOI: 10.1109/CVPR.2019.00342. ISBN 978-1-7281-3294-5.

[163] SAURER, O., BAATZ, G., KÖSER, K., LADICKÝ, L. and POLLEFEYS, M. Image Based Geo-localization in the Alps. *International Journal of Computer Vision*. Berlin, Germany:

Springer Science+Business Media, LLC. 2015, vol. 116, no. 3, p. 213–225. DOI: 10.1007/s11263-015-0830-0. ISSN 0920-5691.

[164] SAXENA, A., CHUNG, S. H. and NG, A. Y. Learning Depth from Single Monocular Images. In: WEISS, Y., SCHÖLKOPF, B. and PLATT, J., ed. *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2006, vol. 18, p. 1161–1168. DOI: 10.1007/s11263-007-0071-y. ISBN 9780262232531.

[165] SAXENA, A., SUN, M. and NG, A. Y. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Los Alamitos, CA, USA: IEEE Computer Society. 2009, vol. 31, no. 5, p. 824–840. DOI: 10.1109/TPAMI.2008.132. ISSN 0162-8828.

[166] SCHINDLER, G., BROWN, M. and SZELISKI, R. City-Scale Location Recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2007, p. 1–7. DOI: 10.1109/CVPR.2007.383150. ISBN 1424411807.

[167] SCHOEFFMANN, K., AHLSTROM, D. and HUDELIST, M. A. 3-D Interfaces to Improve the Performance of Visual Known-Item Search. *IEEE Transactions on Multimedia*. 2014, vol. 16, no. 7, p. 1942–1951. DOI: 10.1109/TMM.2014.2333666. ISSN 1520-9210.

[168] SCHÖNBERGER, J. L., PRICE, T., SATTLER, T., FRAHM, J. and POLLEFEYS, M. A Vote-and-Verify Strategy for Fast Spatial Verification in Image Retrieval. In: LAI, S.-H., LEPETIT, V., NISHINO, K. and SATO, Y., ed. *Computer Vision – ACCV 2016*. Cham, Germany: Springer International Publishing, 2017, vol. 10111, p. 321–337. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-54181-5_21. ISBN 978-3-319-54181-5.

[169] SCHÖNBERGER, J. L., ZHENG, E., FRAHM, J.-M. and POLLEFEYS, M. Pixelwise View Selection for Unstructured Multi-View Stereo. In: LEIBE, B., MATAS, J., SEBE, N. and WELLING, M., ed. *Computer Vision – ECCV 2016*. Cham, Germany: Springer International Publishing, 2016, vol. 9907, p. 501–518. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-46487-9_31. ISBN 978-3-319-46487-9.

[170] SCHÖNBERGER, J. L. and FRAHM, J.-M. Structure-from-Motion Revisited. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2016, p. 4104–4113. DOI: 10.1109/CVPR.2016.445. ISBN 978-1-4673-8852-8.

[171] SEBE, I. O., HU, J., YOU, S. and NEUMANN, U. 3D Video Surveillance with Augmented Virtual Environments. In: *First ACM SIGMM International Workshop on Video Surveillance*. New York, NY, USA: ACM, 2003, p. 107–112. IWVS '03. DOI: 10.1145/982452.982466. ISBN 1-58113-780-X.

[172] SENLET, T., EL GAALY, T. and ELGAMMAL, A. Hierarchical Semantic Hashing: Visual Localization from Buildings on Maps. In: *Proceedings - International Conference on Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2014, p. 2990–2995. DOI: 10.1109/ICPR.2014.516. ISBN 978-1-4799-5209-0.

[173] SEO, P. H., WEYAND, T., SIM, J. and HAN, B. CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps. In: FERRARI, V., HEBERT, M., SMINCHISESCU, C. and WEISS, Y., ed. *Computer Vision – ECCV 2018*. Cham, Germany: Springer International Publishing, 2018, vol. 11214, p. 544–560. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-01249-6_33. ISBN 978-3-030-01249-6.

[174] SEUNGRYONG KIM, DONGBO MIN, BUMSUB HAM, SEUNGCHUL RYU, DO, M. N. et al. DASC: Dense Adaptive Self-Correlation Descriptor for Multi-modal and Multi-spectral Correspondence. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2015, p. 2103–2112. DOI: 10.1109/CVPR.2015.7298822. ISBN 978-1-4673-6964-0.

[175] SHAN, Q., CURLESS, B., FURUKAWA, Y., HERNANDEZ, C. and SEITZ, S. M. Photo Uncrop. In: FLEET, D., PAJDLA, T., SCHIELE, B. and TUYTELAARS, T., ed. *Computer Vision – ECCV 2014*. Cham, Germany: Springer International Publishing, 2014, vol. 8694, p. 16–31. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-10599-4_2. ISBN 978-3-319-10599-4.

[176] SHECHTMAN, E. and IRANI, M. Matching Local Self-Similarities across Images and Videos. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, June 2007, p. 1–8. DOI: 10.1109/CVPR.2007.383198. ISBN 1-4244-1179-3.

[177] SHOTTON, J., GLOCKER, B., ZACH, C., IZADI, S., CRIMINISI, A. et al. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2013, p. 2930–2937. DOI: 10.1109/CVPR.2013.377. ISBN 978-1-5386-5672-3.

[178] SHRIVASTAVA, A., MALISIEWICZ, T., GUPTA, A. and EFROS, A. A. Data-Driven Visual Similarity for Cross-Domain Image Matching. *ACM Transactions on Graphics*.

New York, NY, USA: ACM. 2011, vol. 30, no. 6, p. 1–10. DOI: 10.1145/2070781.2024188. ISSN 0730-0301.

[179] SILBERMAN, N., HOIEM, D., KOHLI, P. and FERGUS, R. Indoor Segmentation and Support Inference from RGBD Image. In: FITZGIBBON, A., LAZEBNIK, S., PERONA, P., SATO, Y. and SCHMID, C., ed. *Computer Vision – ECCV 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7576, p. 746–760. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-33715-4_54. ISBN 978-3-642-33715-4.

[180] SIVIC, J. and ZISSERMAN, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision*. Los Alamitos, CA, USA: IEEE Computer Society, 2003, vol. 2, p. 1470–1477. DOI: 10.1109/ICCV.2003.1238663. ISBN 0-7695-1950-4.

[181] SLATER, M., USOH, M. and STEED, A. Depth of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments*. Cambridge, MA, USA: MIT Press. 1994, vol. 3, no. 2, p. 130–144. DOI: 10.1162/pres.1994.3.2.130. ISSN 1054-7460.

[182] SMITH, S. L. Stopmotion Photowalk Animation for Spatial Immersion in a Remote Cultural Heritage Site. In: *Proceedings of the Conference on Electronic Visualisation and the Arts*. Swindon, UK: BCS Learning & Development Ltd., 2015, p. 298–305. EVA '15. DOI: 10.14236/ewic/eva2015.33.

[183] SNAVELY, N., GARG, R., SEITZ, S. M. and SZELISKI, R. Finding paths through the world's photos. *ACM Transactions on Graphics*. New York, NY, USA: ACM. 2008, vol. 27, no. 3, p. 1. DOI: 10.1145/1360612.1360614. ISSN 0730-0301.

[184] SNAVELY, N., SEITZ, S. M. and SZELISKI, R. Photo tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics*. New York, NY, USA: ACM. 2006, vol. 25, no. 3, p. 835–846. DOI: 10.1145/1141911.1141964. ISSN 0730-0301.

[185] SNAVELY, N., SEITZ, S. M. and SZELISKI, R. Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*. Berlin, Germany: Springer Science+Business Media, LLC. 2008, vol. 80, no. 2, p. 189–210. DOI: 10.1007/s11263-007-0107-3. ISSN 0920-5691.

[186] STEIN, F. and MEDIONI, G. Map-Based Localization Using the Panoramic Horizon. *IEEE Transactions on Robotics and Automation*. New York, NY, USA: IEEE. 1995, vol. 11, p. 892–896. DOI: 10.1109/70.478436. ISSN 1042-296X.

[187] STRABO. *Geographica*. C. 7.

[188] SVÄRM, L., ENQVIST, O., OSKARSSON, M. and KAHL, F. Accurate localization and pose estimation for large 3D models. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2014, p. 532–539. ISBN 9781479951178.

[189] TALLURI, R. and AGGARWAL, J. K. Image Map Correspondence for Mobile Robot Self-Location Using Computer Graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Los Alamitos, CA, USA: IEEE Computer Society. 1993, vol. 15, no. 6, p. 597–601. DOI: 10.1109/34.216729. ISSN 0162-8828.

[190] TALLURI, R. and AGGARWAL, J. Position Estimation for an Autonomous Mobile Robot in an Outdoor Environment. *IEEE Transactions on Robotics and Automation*. New York, NY, USA: IEEE. 1992, vol. 8, no. 5, p. 573–584. DOI: 10.1109/70.163782. ISSN 1042-296X.

[191] THOMEE, B., SHAMMA, D. A., FRIEDLAND, G., ELIZALDE, B., NI, K. et al. YFCC100M: The New Data in Multimedia Research. *Communications of the ACM*. New York, NY, USA: ACM. January 2016, vol. 59, no. 2, p. 64–73. DOI: 10.1145/2812802. ISSN 0001-0782.

[192] TIAN, Y., FAN, B. and WU, F. L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, July 2017, p. 6128–6136. DOI: 10.1109/CVPR.2017.649. ISBN 978-1-5386-0458-8.

[193] TOLIAS, G., AVRITHIS, Y. and JÉGOU, H. Image Search with Selective Match Kernels: Aggregation Across Single and Multiple Images. *International Journal of Computer Vision*. Berlin, Germany: Springer Science+Business Media, LLC. 2016, vol. 116, no. 3, p. 247—–261. DOI: 10.1007/s11263-015-0810-4. ISSN 1573-1405.

[194] TOMASI, C. and KANADE, T. *Detection and Tracking of Point Features* [online]. School of Computer Science, Carnegie Mellon University, 1991. [cit. 2020-12-07]. Available at: https://cecas.clemson.edu/~stb/klt/tomasi-kanade-techreport-1991.pdf.

[195] TOMPKIN, J., PECE, F., SHAH, R., IZADI, S., KAUTZ, J. et al. Video collections in panoramic contexts. In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2013, p. 131–140. UIST '13. DOI: 10.1145/2501988.2502013. ISBN 9781450322683.

[196] TORII, A., ARANDJELOVIĆ, R., SIVIC, J., OKUTOMI, M. and PAJDLA, T. 24/7 place recognition by view synthesis. In: *2015 IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society, 2015, p. 1808–1817. DOI: 10.1109/CVPR.2015.7298790. ISBN 978-1-4673-6964-0.

[197] TORII, A., SIVIC, J., OKUTOMI, M. and PAJDLA, T. Visual Place Recognition with Repetitive Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Los Alamitos, CA, USA: IEEE Computer Society. 2015, vol. 37, no. 11, p. 2346–2359. DOI: 10.1109/TPAMI.2015.2409868. ISSN 0162-8828.

[198] TORNIAI, C., BATTLE, S. and CAYZER, S. Sharing, Discovering and Browsing Geotagged Pictures on the World Wide Web. In: SCHARL, A. and TOCHTERMANN, K., ed. *The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*. London: Springer London, 2007, p. 159–170. DOI: 10.1007/978-1-84628-827-2_15. ISBN 978-1-84628-827-2.

[199] TRIGGS, B., MCLAUCHLAN, P. F., HARTLEY, R. I. and FITZGIBBON, A. W. Bundle Adjustment — A Modern Synthesis. In: TRIGGS, B., ZISSERMAN, A. and SZELISKI, R., ed. *Vision Algorithms: Theory and Practice*. Berlin, Heidelberg: Springer, 2000, p. 298–372. DOI: 10.1007/3-540-44480-7_21. ISBN 978-3-540-44480-0.

[200] TZENG, E., ZHAI, A., CLEMENTS, M., TOWNSHEND, R. and ZAKHOR, A. User-Driven Geolocation of Untagged Desert Imagery Using Digital Elevation Models. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Los Alamitos, CA, USA: IEEE Computer Society, 2013, p. 237–244. DOI: 10.1109/CVPRW.2013.42. ISBN 978-0-7695-4990-3.

[201] UUSITALO, S., ESKOLIN, P., YOU, Y. and BELIMPASAKIS, P. An Extensible Mirror World from User-Generated Content. In: *2010 IEEE Virtual Reality Conference*. New York, NY, USA: IEEE, 2010, p. 311–312. DOI: 10.1109/VR.2010.5444751. ISBN 978-1-4244-6237-7.

[202] VACA CASTANO, G., ZAMIR, A. R. and SHAH, M. City Scale Geo-Spatial Trajectory Estimation of a Moving Camera. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2012, p. 1186–1193. DOI: 10.1109/CVPR.2012.6247800. ISBN 978-1-4673-1226-4.

[203] VEAS, E., MULLONI, A., KRUIJFF, E., SCHMALSTIEG, D., REGENBRECHT, H. et al. Techniques for View Transition in Multi-Camera Outdoor Environments. In: *Proceedings of Graphics Interface*. Toronto, Ontario, Canada: Canadian Information Processing Society, 2010, p. 193–200. DOI: 10.5555/1839214.1839248. ISBN 978-1-56881-712-5. Available at: http://dl.acm.org/citation.cfm?id=1839214.1839248.

[204] VINCENTY, T. Direct and Inverse Solutions of Geodetics on the Ellipsoid with Application of Nested Equations. *Survey Review*. Taylor & Francis. 1975, vol. 23, no. 176, p. 88–93. DOI: 10.1179/sre.1975.23.176.88. ISSN 0039-6265.

[205] VIOLA, P. and WELLS, W. M. Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*. Berlin, Germany: Springer Science+Business Media, LLC. 1997, vol. 24, no. 2, p. 137–154. DOI: 10.1023/A:1007958904918. ISSN 09205691.

[206] VISWANATHAN, A., PIRES, B. R. and HUBER, D. Vision Based Robot Localization by Ground to Satellite Matching in GPS-denied Situations. In: *IEEE International Conference on Intelligent Robots and Systems*. New York, NY, USA: IEEE, 2014, p. 192–198. DOI: 10.1109/IROS.2014.6942560. ISBN 9781479969340.

[207] VO, N., JACOBS, N. and HAYS, J. Revisiting IM2GPS in the Deep Learning Era. In: *Proceedings of the IEEE International Conference on Computer Vision*. Los Alamitos, CA, USA: IEEE Computer Society, 2017. DOI: 10.1109/ICCV.2017.286. ISBN 978-1-5386-1033-6.

[208] WANG, C.-P., WILSON, K. and SNAVELY, N. Accurate Georegistration of Point Clouds Using Geographic Data. In: *2013 International Conference on 3D Vision – 3DV 2013*. Los Alamitos, CA, USA: IEEE Computer Society, 2013, p. 33–40. DOI: 10.1109/3DV.2013.13. ISBN 978-0-7695-5067-1.

[209] WANG, M., LIANG, J. B., ZHANG, S. H., LU, S. P., SHAMIR, A. et al. Hyper-Lapse From Multiple Spatially-Overlapping Videos. *IEEE Transactions on Image Processing*. IEEE. April 2018, vol. 27, no. 4, p. 1735–1747. DOI: 10.1109/TIP.2017.2749143. ISSN 1057-7149.

[210] WANG, P., YANG, R., CAO, B., XU, W. and LIN, Y. DeLS-3D: Deep Localization and Segmentation with a 3D Semantic Map. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2018, p. 5860–5869. DOI: 10.1109/CVPR.2018.00614. ISBN 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[211] WANG, Y., BOWMAN, D., KRUM, D., COALHO, E., SMITH JACKSON, T. et al. Effects of Video Placement and Spatial Context Presentation on Path Reconstruction Tasks with Contextualized Videos. *IEEE Transactions on Visualization and Computer Graphics*. Los Alamitos, CA, USA: IEEE Computer Society. November 2008, vol. 14, no. 6, p. 1755–1762. DOI: 10.1109/TVCG.2008.126. ISSN 1077-2626.

[212] WANG, Y., KRUM, D. M., COELHO, E. M. and BOWMAN, D. A. Contextualized Videos: Combining Videos with Environment Models to Support Situational Understanding. *IEEE Transactions on Visualization and Computer Graphics*. Los Alamitos, CA, USA: IEEE Computer Society. November 2007, vol. 13, no. 6, p. 1568–1575. DOI: 10.1109/TVCG.2007.70544. ISSN 1077-2626.

[213] WEYAND, T., KOSTRIKOV, I. and PHILBIN, J. PlaNet - Photo Geolocation with Convolutional Neural Networks. In: LEIBE, B., MATAS, J., SEBE, N. and WELLING, M., ed. *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*. Cham, Germany: Springer International Publishing, 2016, vol. 9912, p. 37–55. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-46484-8_3. ISBN 978-3-319-46484-8.

[214] WOO, J., KWEON, I., KIM, G. and KIM, I. Robust horizon and peak extraction for vision-based navigation. In: *Proceedings of the IAPR Conference on Machine Vision Applications*. 2005, p. 526–529. DOI: 10.1.1.144.3983. ISBN 4901122045.

[215] WOO, J., SON, K., LI, T., KIM, G. S. and KWEON, I.-S. Vision-based UAV Navigation in Mountain Area. In: *Proceedings of the IAPR Conference on Machine Vision Applications*. 2007, p. 236–239. ISBN 978-4-901122-07-8.

[216] WORKMAN, S., SOUVENIR, R. and JACOBS, N. Wide-Area Image Geolocalization with Aerial Reference Imagery. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision*. New York, NY, USA: IEEE, 2015, p. 3961–3969. DOI: 10.1109/ICCV.2015.451. ISBN 978-1-4673-8391-2.

[217] WU, C. Towards Linear-Time Incremental Structure from Motion. In: *International Conference on 3D Vision*. Los Alamitos, CA, USA: IEEE Computer Society, June 2013, p. 127–134. DOI: 10.1109/3DV.2013.25. ISBN 978-0-7695-5067-1.

[218] WU, C., AGARWAL, S., CURLESS, B. and SEITZ, S. M. Multicore Bundle Adjustment. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, June 2011, p. 3057–3064. DOI: 10.1109/CVPR.2011.5995552. ISBN 978-1-4577-0395-9.

[219] WU, F. and TORY, M. PhotoScope: Visualizing Spatiotemporal Coverage of Photos for Construction Management. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009, p. 1103–1112. CHI '09. DOI: 10.1145/1518701.1518869. ISBN 978-1-60558-246-7.

[220] YANG, C. and MEDIONI, G. Object Modelling by Registration of Multiple Range Images. *Image and Vision Computing*. Oxford, UK: Butterworth-Heinemann. 1992, vol. 10, no. 3, p. 145–155. DOI: 10.1016/0262-8856(92)90066-C. ISSN 0262-8856.

[221] YI, K. M., TRULLS, E., LEPETIT, V. and FUA, P. LIFT: Learned Invariant Feature Transform. In: LEIBE, B., MATAS, J., SEBE, N. and WELLING, M., ed. *Computer Vision – ECCV 2016*. Cham, Germany: Springer International Publishing, 2016, p. 467–483. DOI: 10.1007/978-3-319-46466-4_28. ISBN 978-3-319-46466-4.

[222] YOU, R.-J. Transformation of Cartesian to Geodetic Coordinates without Iterations. *Journal of Surveying Engineering*. ASCE. 2000, vol. 126, no. 1, p. 1–7. DOI: 10.1061/(ASCE)0733-9453(2000)126:1(1). ISSN 0733-9453.

[223] YU, F., CHEN, H., WANG, X., XIAN, W., CHEN, Y. et al. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2020, p. 2633–2642. DOI: 10.1109/CVPR42600.2020.00271. ISBN 978-1-7281-7169-2.

[224] ZAMIR, A. R., ARDESHIR, S. and SHAH, M. GPS-Tag Refinement Using Random Walks with an Adaptive Damping Factor. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2014, p. 4280–4287. DOI: 10.1109/CVPR.2014.545. ISBN 978-1-4799-5118-5.

[225] ZAMIR, A. R. and SHAH, M. Accurate Image Localization Based on Google Maps Street View. In: DANIILIDIS, K., MARAGOS, P. and PARAGIOS, N., ed. *Computer Vision – ECCV 2010*. Berlin, Heidelberg: Springer, 2010, vol. 6314, p. 255–268. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-15561-1_19. ISBN 978-3-642-15561-1.

[226] ZAMIR, A. R. and SHAH, M. Image Geo-Localization Based on Multiple Nearest Neighbor Feature Matching Using Generalized Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Los Alamitos, CA, USA: IEEE Computer Society. 2014, vol. 36, no. 8, p. 1546–1558. DOI: 10.1109/TPAMI.2014.2299799. ISSN 0162-8828.

[227] ZEISL, B., SATTLER, T. and POLLEFEYS, M. Camera Pose Voting for Large-Scale Image-Based Localization. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, 2015, p. 2704–2712. DOI: 10.1109/ICCV.2015.310. ISBN 978-1-4673-8391-2.

[228] ZHANG, W. and KOŠECKÁ, J. Image Based Localization in Urban Environments. In: *Third International Symposium on 3D Data Processing, Visualization, and Transmission*

(3DPVT'06). Los Alamitos, CA, USA: IEEE Computer Society, 2006, p. 33–40. DOI: 10.1109/3DPVT.2006.80. ISBN 0-7695-2825-2.

[229] ZHANG, Z. Determining the Epipolar Geometry and its Uncertainty: A Review. *International Journal of Computer Vision*. Berlin, Germany: Springer Science+Business Media, LLC. 1998, vol. 27, no. 2, p. 161–195. DOI: 10.1023/a:1007941100561. ISSN 0920-5691.

[230] ZHENG, Y. T., ZHAO, M., SONG, Y., ADAM, H., BUDDEMEIER, U. et al. Tour the World: Building a web-scale landmark recognition engine. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, p. 1085–1092. DOI: 10.1109/CVPR.2009.5206749. ISBN 978-1-4244-3992-8.

[231] ZHOU, B., LAPEDRIZA, A., KHOSLA, A., OLIVA, A. and TORRALBA, A. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Los Alamitos, CA, USA: IEEE Computer Society. 2018, vol. 40, no. 6, p. 1452–1464. DOI: 10.1109/TPAMI.2017.2723009. ISSN 0162-8828.

[232] ZHOU, B., LAPEDRIZA, A., XIAO, J., TORRALBA, A. and OLIVA, A. Learning Deep Features for Scene Recognition using Places Database. In: GHAHRAMANI, Z., WELLING, M., CORTES, C., LAWRENCE, N. and WEINBERGER, K. Q., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014, vol. 27, p. 487–495.

[233] ZHU, J. Conversion of Earth-Centered Earth-Fixed Coordinates to Geodetic Coordinates. *IEEE Transactions on Aerospace and Electronic Systems*. New York, NY, USA: IEEE. 1994, vol. 30, no. 3, p. 957–961. DOI: 10.1109/7.303772. ISSN 0018-9251.

[234] ZHU, Z., HUANG, H. Z., TAN, Z. P., XU, K. and HU, S. M. Faithful Completion of Images of Scenic Landmarks Using Internet Images. *IEEE Transactions on Visualization and Computer Graphics*. Los Alamitos, CA, USA: IEEE Computer Society. 2016, vol. 22, no. 8, p. 1945–1958. DOI: 10.1109/TVCG.2015.2480081. ISSN 10772626.

Publications

My work at FIT BUT and Adobe Research led to the following publications, forming the backbone of this theis, listed chronologically.

[A1] Brejcha, J. and Čadík, M. GeoPose3K: Mountain landscape dataset for camera pose estimation in outdoor environments. *Image and Vision Computing*. 2017, vol. 66, p. 1–14. DOI: 10.1016/j.imavis.2017.05.009. ISSN 0262-8856.

[A2] Brejcha, J. and Čadík, M. State-of-the-Art in Visual Geo-Localization. *Pattern Analysis and Applications*. Berlin, Heidelberg: Springer-Verlag. 2017, vol. 20, no. 3, p. 613–637. DOI: 10.1007/s10044-017-0611-1. ISSN 1433-7541.

[A3] Brejcha, J. and Čadík, M. Camera Orientation Estimation in Natural Scenes Using Semantic Cues. In: *2018 International Conference on 3D Vision (3DV)*. New York, NY, USA: IEEE, 2018, p. 208–217. DOI: 10.1109/3DV.2018.00033. ISBN 978-1-5386-8425-2.

[A4] Brejcha, J., Lukáč, M., Chen, Z., DiVerdi, S. and Čadík, M. Immersive Trip Reports. In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2018, p. 389–401. DOI: 10.1145/3242587.3242653. ISBN 9781450359481.

[A5] Brejcha, J., Lukáč, M., Hold Geoffroy, Y., Wang, O. and Čadík, M. LandscapeAR: Large Scale Outdoor Augmented Reality by Matching Photographs with Terrain Models Using Learned Descriptors. In: Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.-M., ed. *Computer Vision – ECCV 2020*. Cham, Germany: Springer International Publishing, 2020, vol. 12374, p. 295–312. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-58526-6_18. ISBN 978-3-030-58525-9.

[A6] Lukáč, M., Chen, Z., Brejcha, J. and Čadík, M. *Generating Immersive Trip Photograph Visualizations* [US Patent Application 16/144,487]. 2020.

Appendix A

Cameras in GeoPose3K

We present a complete list of cameras from the GeoPose3K dataset in Tab. A.1.

Camera model	#	Camera model	#
Canon DIGITAL IXUS 860 IS	253	Canon EOS 650D	2
Canon EOS 6D	112	Canon EOS DIGITAL REBEL XTi	2
COOLPIX L5	109	Canon EOS-1D Mark II N	2
Canon PowerShot G9	90	Canon EOS-1D Mark III	2
iPhone 5	89	Canon PowerShot A530	2
NIKON D300	76	Canon PowerShot S3 IS	2
iPhone 4	72	Canon PowerShot S60	2
NIKON D80	63	Canon PowerShot SX120 IS	2
NIKON D7000	60	Canon PowerShot SX200 IS	2
DMC-TZ5	59	Canon PowerShot SX260 HS	2
NIKON D90	59	DMC-FS10	2
Canon EOS 400D DIGITAL	54	DMC-FT1	2
Canon PowerShot D10	52	DMC-FZ28	2
EX-S600	46	DMC-FZ62	2
Canon DIGITAL IXUS 870 IS	45	DMC-GF1	2
NIKON D700	45	DMC-GF2	2
Canon EOS 450D	42	DMC-TS2	2
Canon EOS 7D	39	DMC-TZ4	2
SLT-A55V	37	DMC-TZ41	2
Canon DIGITAL IXUS 970 IS	36	DSLR-A700	2
SLT-A77V	34	DiMAGE A1	2
Canon EOS 60D	30	DiMAGE A2	2
Canon PowerShot G11	30	DiMAGE Z1	2
NIKON D800	30	Digimax V5 / Kenox V5	2
NIKON D3X	29	E5900	2
Canon EOS 350D DIGITAL	28	E7900	2
Canon PowerShot S95	27	EX-Z120	2
Canon EOS 5D Mark II	26	EX-Z5	2
DMC-TZ10	26	EX-Z700	2
NIKON D40	26	FinePix S2000HD	2
Canon EOS 40D	25	KODAK DX4530 ZOOM DIGITAL CAMERA	2
NIKON D5000	25	KODAK EASYSHARE V1273 DIGITAL CAMERA	2
NIKON D60	25	KODAK Z612 ZOOM DIGITAL CAMERA	2
DSC-RX100	23	NEX-5	2
Canon EOS 500D	22	NIKON D3200	2
DSLR-A290	22	NIKON D3S	2
Canon DIGITAL IXUS 40	21	PENTAX DL	2
Canon EOS 5D Mark III	21	PENTAX K-7	2
Canon PowerShot S100	20	PENTAX Optio VS20	2
iPhone 4S	20	TG-1	2
Canon PowerShot A710 IS	19	iPad	2
E-P3	18	iPhone	2
M9 Digital Camera	18	iPhone 3GS	2
NEX-7	18	Digimax U-CA 5, Kenox U-CA 5 / Kenox U-CA 50	1
Canon EOS 50D	17	KENOX S860 / Samsung S860	1
Canon EOS REBEL T3	17	C-5000Z	1
Canon PowerShot G10	17	C40Z,D40Z	1
PENTAX K100D	17	C720UZ	1
Canon DIGITAL IXUS 800 IS	15	COOLPIX AW110	1
Canon EOS 1000D	15	COOLPIX L1	1

Canon EOS 550D	15	COOLPIX L27	1
Canon PowerShot A640	15	COOLPIX P1	1
COOLPIX AW100	14	COOLPIX P300	1
KODAK EASYSHARE Z950 DIGITAL CAMERA	13	COOLPIX P510	1
NIKON D200	13	COOLPIX P7000	1
NIKON D800E	13	COOLPIX P7700	1
DMC-FZ38	12	COOLPIX P80	1
DiMAGE 7i	12	COOLPIX S10	1
SLT-A65V	12	COOLPIX S230	1
Canon EOS 20D	11	COOLPIX S500	1
DiMAGE A200	11	COOLPIX S610	1
DiMAGE Z5	11	Canon DIGITAL IXUS 330	1
DMC-LX3	10	Canon DIGITAL IXUS 500	1
DSLR-A550	10	Canon DIGITAL IXUS 700	1
E4500	10	Canon DIGITAL IXUS 750	1
NEX-3	10	Canon DIGITAL IXUS 90 IS	1
NIKON D3100	10	Canon EOS REBEL T2i	1
Canon EOS 30D	9	Canon EOS-1Ds Mark II	1
DMC-TZ35	9	Canon IXY DIGITAL 25 IS	1
DSLR-A350	9	Canon PowerShot A3200 IS	1
NIKON D5100	9	Canon PowerShot A4000 IS	1
NIKON D70	9	Canon PowerShot A470	1
PENTAX K200D	9	Canon PowerShot A590 IS	1
COOLPIX P520	8	Canon PowerShot A610	1
DSLR-A900	8	Canon PowerShot A700	1
PENTAX K-3	8	Canon PowerShot A80	1
PENTAX K10D	8	Canon PowerShot A95	1
PENTAX Optio 33L	8	Canon PowerShot G1 X	1
S1	8	Canon PowerShot G5	1
Canon EOS 300D DIGITAL	7	Canon PowerShot G7	1
Canon PowerShot G12	7	Canon PowerShot Pro1	1
DMC-FZ18	7	Canon PowerShot S30	1
DMC-TZ20	7	Canon PowerShot S50	1
DYNAX 7D	7	Canon PowerShot S80	1
NEX-6	7	Canon PowerShot SD1000	1
X-E1	7	Canon PowerShot SD700 IS	1
COOLPIX S9100	6	Canon PowerShot SX100 IS	1
Canon DIGITAL IXUS 55	6	Canon PowerShot SX220 HS	1
Canon EOS 600D	6	D40	1
Canon PowerShot A650 IS	6	D700	1
FinePix S5600	6	DMC-FX37	1
NIKON D4	6	DMC-FX40	1
NIKON D7100	6	DMC-FX8	1
Digimax S1000 / Kenox S1000	5	DMC-FZ30	1
COOLPIX L22	5	DMC-FZ5	1
COOLPIX P5000	5	DMC-FZ50	1
COOLPIX P6000	5	DMC-FZ7	1
COOLPIX S620	5	DMC-G2	1
Canon EOS 5D	5	DMC-LS75	1
Canon EOS DIGITAL REBEL XSi	5	DMC-LX5	1
Canon PowerShot A720 IS	5	DMC-TZ15	1
DMC-GH2	5	DMC-TZ3	1
DMC-TZ18	5	DMC-TZ7	1
DSLR-A500	5	DSC-W120	1
E-M5	5	DSLR-A230	1
PENTAX K-5	5	DSLR-A580	1
PENTAX Optio W20	5	DiMAGE G500	1
VSCOcam	5	DiMAGE X1	1
iPhone 5s	5	E-510	1
Canon DIGITAL IXUS 850 IS	4	E3100	1
Canon EOS REBEL T1i	4	E3500	1
Canon PowerShot S2 IS	4	E4600	1
DMC-FT3	4	E5200	1
DMC-FX01	4	EOS 40D	1
DSLR-A200	4	EX-FH20	1
FinePix F31fd	4	EX-H20G	1
FinePix2800ZOOM	4	EX-Z110	1

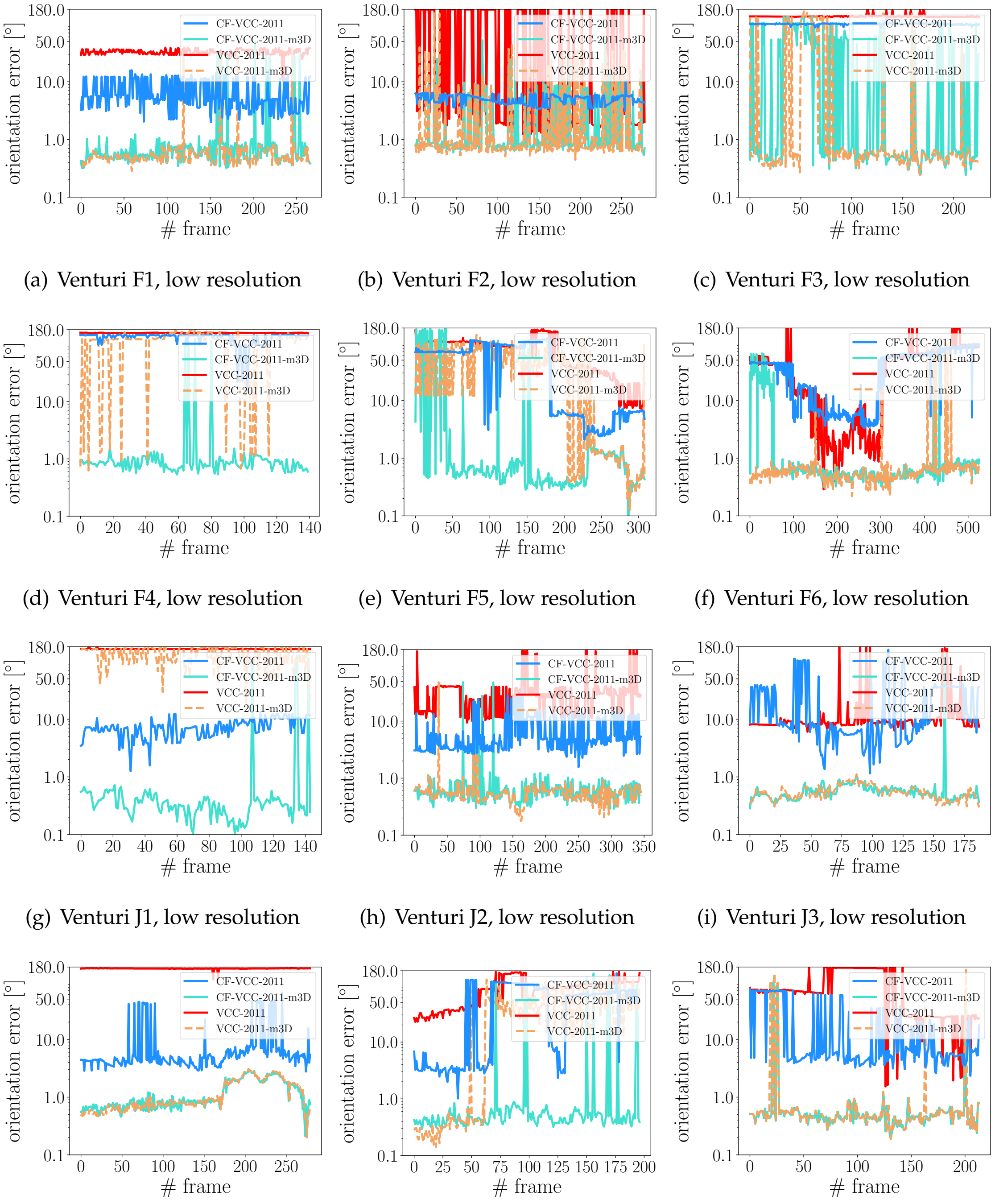
KODAK EASYSHARE C613 ZOOM DIGITAL CAMERA	4	EX-Z4	1
N97	4	EX-Z40	1
NIKON D300S	4	EX-Z55	1
NIKON D50	4	EX-Z60	1
NIKON D70s	4	EX-Z750	1
PENTAX DS	4	FinePix A800	1
SAMSUNG WB550, WB560 / VLUU WB550 / SAMSUNG HZ15W	4	FinePix F30	1
Digimax S830 / Kenox S830	3	FinePix F450	1
COOLPIX P5100	3	FinePix J150W	1
COOLPIX S4	3	FinePix S200EXR	1
Canon DIGITAL IXUS 65	3	FinePix S5000	1
Canon DIGITAL IXUS 950 IS	3	FinePix S6500fd	1
Canon EOS 1100D	3	FinePix S7000	1
Canon PowerShot A620	3	HP PhotoSmart C945 (V01.46)	1
Canon PowerShot A70	3	HP PhotoSmart R707 (V01.00)	1
Canon PowerShot S5 IS	3	KODAK CX7530 ZOOM DIGITAL CAMERA	1
Canon PowerShot SX230 HS	3	KODAK DX7440 ZOOM DIGITAL CAMERA	1
DC P500	3	KODAK EASYSHARE C195 Digital Camera	1
DMC-FX35	3	KODAK EASYSHARE ZD710 ZOOM DIGITAL CAMERA	1
DMC-FZ200	3	KODAK V610 DUAL LENS DIGITAL CAMERA	1
DMC-G3	3	Konica Digital Camera KD-400Z	1
DMC-TZ30	3	LEICA X1	1
DMC-TZ31	3	NEX-3N	1
DMC-TZ6	3	NIKON D100	1
DSLR-A100	3	NIKON D1X	1
DSLR-A300	3	NIKON D2Xs	1
E-PL3	3	NIKON D3000	1
EX-Z75	3	NV20, VLUU NV20	1
FinePix JZ500	3	PENTAX DL2	1
Hasselblad H3D	3	PENTAX K-r	1
KODAK Z740 ZOOM DIGITAL CAMERA	3	PENTAX K100D Super	1
NIKON D3	3	PENTAX Optio S4	1
NIKON D40X	3	PENTAX Optio S7	1
NIKON D600	3	PENTAX Optio WPi	1
PENTAX K-x	3	QV-R52	1
VR330,D730	3	SAMSUNG ES15 / VLUU ES15 / SAMSUNG SL30	1
WB2000	3	SAMSUNG ES55,ES57 / VLUU ES55 / SAMSUNG SL102	1
C750UZ	2	SAMSUNG ES74,ES75,ES78 / VLUU ES75,ES78	1
COOLPIX S9500	2	SAMSUNG WB850F/WB855F	1
Canon DIGITAL IXUS 50	2	SLT-A35	1
Canon DIGITAL IXUS 70	2	SLT-A57	1
Canon DIGITAL IXUS 80 IS	2	SLT-A99V	1
Canon DIGITAL IXUS 85 IS	2	SP560UZ	1
Canon EOS 100D	2	X-Pro1	1

Table A.1: Number of images per camera model in the *GeoPose3K* dataset.

Appendix B

Additional Experiments on Venturi Mountain Dataset

We provide complete results on Venturi Mountain dataset [142] generated by our method using Confidence Fusion (CF) with semantic segments discussed in Chapter 6. We plot camera orientation error per frame, on both low (Fig. B.1) and high (Fig. B.2) resolution. We also present results for all discussed variants of our CF method and keypoint-based method for camera pose estimation (LSAR-Ours) presented in Chapter 7 on Venturi mountain dataset in Tab. B.1, including mean and median of the orientation error for each sequence.



(a) Venturi F1, low resolution

(b) Venturi F2, low resolution

(c) Venturi F3, low resolution

(d) Venturi F4, low resolution

(e) Venturi F5, low resolution

(f) Venturi F6, low resolution

(g) Venturi J1, low resolution

(h) Venturi J2, low resolution

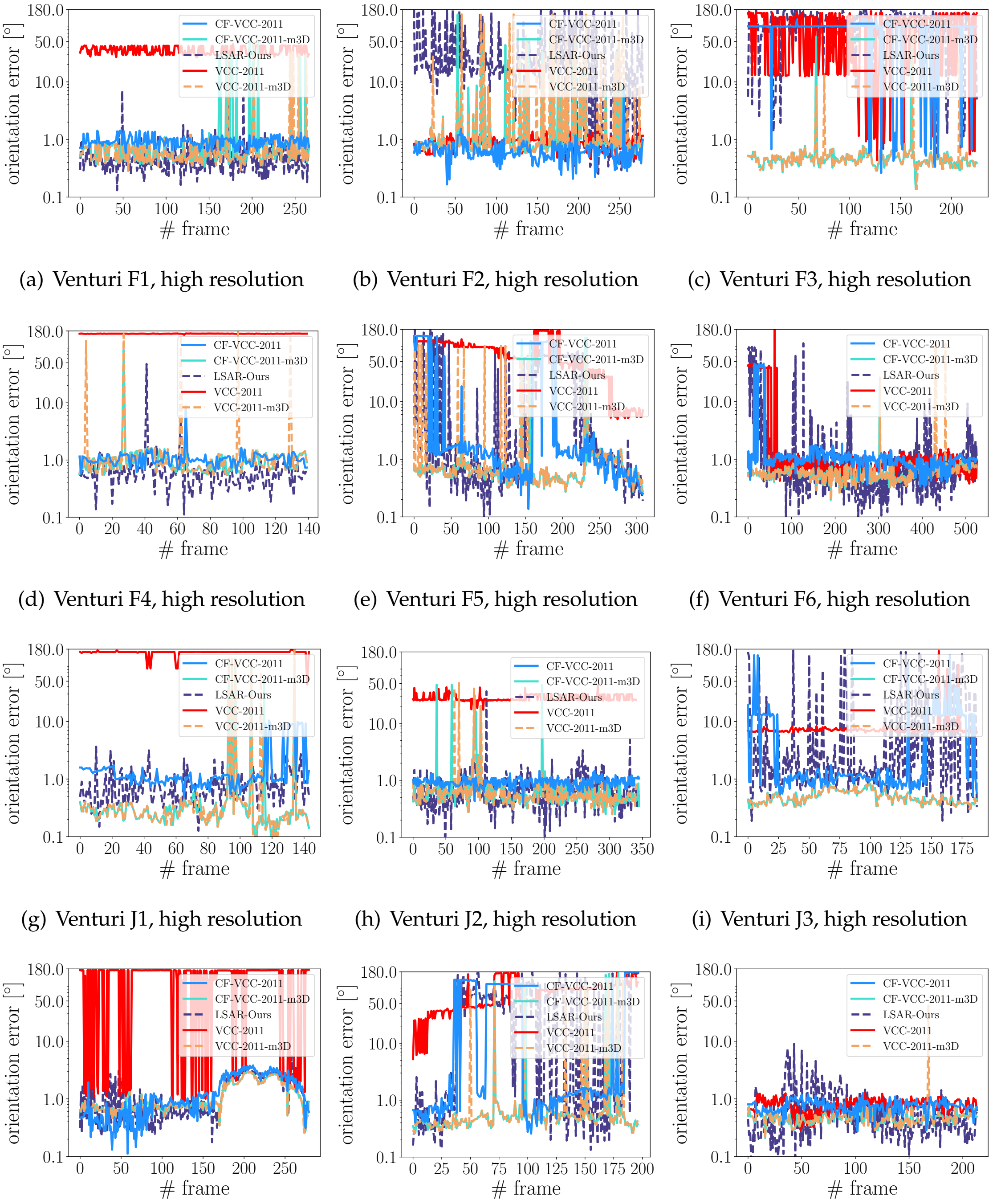
(i) Venturi J3, low resolution

(j) Venturi J4, low resolution

(k) Venturi J5, low resolution

(l) Venturi J6, low resolution

Figure B.1: Per-frame orientation error on Venturi mountain dataset (low resolution).



(a) Venturi F1, high resolution

(b) Venturi F2, high resolution

(c) Venturi F3, high resolution

(d) Venturi F4, high resolution

(e) Venturi F5, high resolution

(f) Venturi F6, high resolution

(g) Venturi J1, high resolution

(h) Venturi J2, high resolution

(i) Venturi J3, high resolution

(j) Venturi J4, high resolution

(k) Venturi J5, high resolution

(l) Venturi J6, high resolution

Figure B.2: Per-frame orientation error on Venturi mountain dataset (high resolution).

Resolution	Method	Avg. mean	Avg. stddev	F1		F2		F3		F4		F5		F6	
				mean	med.	mean	med.	mean	med.	mean	med.	mean	med.	mean	med.
low	**CF-VCC-2011-m3D**	**5.93**	**21.82**	1.82	0.55	**3.50**	0.82	30.26	0.59	**4.15**	**0.83**	**13.92**	**0.58**	4.02	0.58
	VCC-2011-m3D	21.06	44.20	**1.00**	**0.49**	6.01	**0.72**	**21.27**	**0.53**	116.87	122.63	41.30	38.91	**1.69**	**0.56**
	CF-VCC-2011	34.19	41.75	6.67	5.08	5.00	5.09	100.11	100.57	132.06	142.88	51.42	68.55	39.95	44.38
	VCC-2011	79.94	64.55	32.68	32.45	72.36	2.23	136.30	134.63	157.80	157.78	74.74	85.07	44.36	42.53
	CF	64.25	59.63	5.43	5.67	68.89	6.20	96.89	97.17	50.88	17.16	84.85	66.95	97.57	120.97
high	**CF-VCC-2011-m3D**	**1.92**	**10.62**	2.57	0.56	3.68	0.78	**1.06**	**0.45**	1.57	0.82	**2.68**	**0.60**	**0.61**	0.57
	VCC-2011-m3D	2.88	14.72	1.49	0.55	8.94	0.79	1.27	**0.45**	6.25	0.86	4.42	**0.60**	1.18	0.57
	CF-VCC-2011	12.42	32.44	0.93	0.90	**0.67**	**0.63**	85.68	91.73	1.09	1.01	21.18	1.22	2.45	0.99
	VCC-2011	52.44	62.36	35.21	32.45	0.93	0.89	86.27	131.80	161.23	161.34	70.05	57.61	5.37	0.84
	CF	60.40	61.97	1.69	1.58	85.50	140.40	96.87	96.88	60.76	18.15	77.89	59.93	90.48	114.20
-	HLoc-synthetic	28.00	50.54	52.73	3.11	1.84	0.42	11.54	0.44	36.08	11.51	4.17	1.55	10.21	0.83
-	HLoc-deeplab	98.76	61.24	133.69	143.40	47.52	2.17	85.66	89.11	128.47	141.65	54.48	20.21	120.40	128.22
-	LSAR-Ours	19.2	41.86	**0.62**	**0.44**	40.92	17.08	104.53	110.48	**1.03**	**0.59**	23.01	1.18	3.79	**0.56**

Resolution	Method			J1		J2		J3		J4		J5		J6	
				mean	med.	mean	med.	mean	med.	mean	med.	mean	med.	mean	med.
low	**CF-VCC-2011-m3D**			**3.51**	**0.33**	1.20	0.57	1.31	**0.51**	**1.20**	0.86	**5.93**	**0.46**	**2.41**	**0.47**
	VCC-2011-m3D			132.92	162.81	**0.71**	**0.53**	**0.55**	0.54	**1.20**	**0.81**	41.04	36.50	4.46	**0.47**
	CF-VCC-2011			7.41	7.17	6.75	4.57	23.87	8.54	8.85	4.68	55.39	68.35	19.01	5.41
	VCC-2011			165.12	164.99	34.25	28.30	15.96	8.53	131.28	169.19	86.24	87.79	81.79	67.31
	CF			4.71	4.57	12.64	4.67	92.17	139.49	55.97	55.36	128.39	123.76	45.01	29.43
high	**CF-VCC-2011-m3D**			4.54	**0.26**	1.26	0.54	**0.50**	**0.47**	**1.18**	**0.80**	**5.24**	**0.47**	**0.47**	0.47
	VCC-2011-m3D			5.17	**0.26**	1.08	**0.53**	**0.50**	**0.47**	**1.18**	**0.80**	6.29	**0.47**	0.66	0.47
	CF-VCC-2011			1.85	1.05	0.93	0.93	8.32	1.13	1.42	1.02	41.65	1.29	0.75	0.76
	VCC-2011			158.06	160.73	27.77	25.93	8.75	6.90	132.92	169.56	79.56	85.01	0.78	0.78
	CF			2.68	2.67	7.14	1.48	79.02	138.09	57.24	51.70	132.40	128.91	9.60	3.73
-	HLoc-synthetic			115.54	116.50	86.08	114.19	6.01	1.83	4.11	1.13	3.84	1.54	40.85	0.48
-	HLoc-deeplab			115.23	116.59	134.89	142.98	28.61	3.10	100.10	161.00	57.67	14.72	155.35	155.07
-	LSAR-Ours			**0.85**	0.64	**0.84**	0.61	19.21	1.61	1.41	0.9	54.62	50.24	0.8	**0.42**

Table B.1: Mean and median orientation error (in degrees) of the Confidence Fusion (CF) proposed in Chapter 6, and the keypoint-based camera pose estimation method (LSAR-Ours) introduced in Chapter 7 on Venturi Mountain dataset.