Ground-to-Aerial Image Matching for Vehicle Localization

Xia, Z.

DOI
10.4233/uuid:3f52a2eb-abc2-478c-a534-28c661953f0d

Publication date
2024

Ground-to-Aerial Image Matching for Vehicle Localization

Ground-to-Aerial Image Matching for Vehicle Localization

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Wednesday, 11 December 2024 at 10:00 o'clock

by

Zimin XIA

This dissertation has been approved by the promotors

Composition of the doctoral committee:

Rector Magnificus,	Chairperson
Dr. Julian F. P. Kooij,	Delft University of Technology, promotor
Prof. dr. Dariu M. Gavrila,	Delft University of Technology, promotor
Independent members:	
Prof. dr. Guido de Croon	Delft University of Technology
Dr. Victor A. Prisacariu	University of Oxford, The United Kingdom
Dr. Martin R. Oswald	University of Amsterdam, The Netherlands
Dr. Jan van Gemert	Delft University of Technology
Prof. dr. Jantien E. Stoter	Delft University of Technology

Keywords: intelligent vehicles, aerial imagery, visual localization

Style: TU Delft House Style, with modifications by Moritz Beller
`https://github.com/Inventitech/phd-thesis-template`

ISBN: 978-94-6384-691-2

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

All things are difficult before they are easy.

Dr. Thomas Fuller

Contents

Summary

Automated driving has immense potential for improving road safety. Over the past decades, extensive research has been conducted in this field. Although the technological capability for highly automated driving exists today, its widespread application is not yet present. One major limiting factor of current automated driving solutions is that vehicle localization heavily relies on high-definition maps (HD maps), which are highly expensive to construct and maintain.

This dissertation focuses on developing a more scalable solution for vehicle localization. It explores a novel technique that estimates the ego vehicle's pose (location and orientation) by matching ground-level images captured by the vehicle's onboard camera to publicly available geo-referenced aerial imagery. Specifically, Chapter 1 begins with a brief motivation for this ground-to-aerial cross-view visual localization task. Then it defines the main and sub-research questions of this dissertation. Following this, Chapter 2 reviews the literature relevant to cross-view visual localization. Then, subsequent chapters comprehensively address cross-view visual localization from various perspectives and answer the research questions at the end.

Chapter 3 identifies a limitation in existing image retrieval-based cross-view localization methods: these models are trained for large-scale coarse localization rather than accurate localization in geographical local areas. This formulation is not suitable for automated driving, as the coarse location of the vehicle is often known from GNSS or temporal information, while the accurate location within a small local area is more crucial. Therefore, Chapter 3 proposes to incorporate the coarse localization prior from other sensors, such as GNSS, into the training of the image retrieval model to force the model to learn locally discriminative features. It achieves this by introducing a novel loss function, the Geo-local Triplet Loss, and demonstrates the effectiveness and generality of this loss with two baseline models on two datasets. Furthermore, a commonly used temporal filtering pipeline is implemented to validate that fusing the model, trained with the proposed loss, with GNSS positioning yields better localization accuracy than either combining the baseline with GNSS or using GNSS alone.

Next, Chapter 4 identifies the limitations of using image retrieval for accurate vehicle localization, that better localization requires increased computation and storage as reference aerial image patches must densely cover the target area. Therefore, instead of formulating the localization problem as image retrieval, Chapter 4 proposes to directly match the ground-level image to a known aerial image covering the local surroundings for vehicle pose estimation (localization plus orientation estimation). It introduces a novel method called Convolutional Cross-View Pose Estimation (CCVPE), which constructs orientation-aware ground and aerial image descriptors for the joint estimation of the location and orientation of the ground-level camera. Experiments show that CCVPE achieves state-of-the-art accuracy in localization and orientation estimation on three cross-view localization benchmarks.

Afterward, Chapter 5 focuses on improving the runtime efficiency of cross-view localization. It proposes SliceMatch, a generative-testing method that constructs orientation-aware ground and aerial image descriptors by explicitly utilizing the geometric relationship between ground and aerial views. SliceMatch uses pre-computed ground and aerial slice masks to guide the feature aggregation when constructing the ground and aerial descriptors. Since the slice masks can be computed in advance, and the feature masking and descriptor comparison can be implemented as matrix multiplication, a highly parallelizable process, SliceMatch achieves a fast runtime, of over 167 FPS on the VIGOR dataset.

The scalability of cross-view localization is studied in Chapter 6. In the envisioned application, a cross-view localization model would be trained in areas where ground truth data is available, and then deployed in other areas where there is no ground truth. Due to the domain gap between the training and test regions, such direct generalization always leads to a performance drop. In practice, even though acquiring accurate ground truth location data is expensive, requiring mobile mapping vehicles with expensive sensor kits, collecting ground-level images with coarse ground truth, which may have errors of tens of meters, is easier, such as using mobile phones and their built-in GNSS. Hence, Chapter 6 proposes to leverage the easy-to-collect coarse ground truth data for weakly-supervised fine-tuning of pre-trained cross-view localization models. The coarse ground truth data is used to pair the ground image in the target area with an aerial image covering its local surroundings. Chapter 6 introduces a knowledge self-distillation framework that uses a pre-trained model as a teacher to generate pseudo ground truth for each ground-aerial image pair. The noise in the pseudo ground truth is suppressed, and the large outliers are removed using the proposed techniques. Subsequently, a student model is trained using the improved pseudo ground truth. Experiments with two baselines on two benchmarks demonstrate that the proposed weakly-supervised knowledge self-distillation can lead up to 20% accuracy gain.

Finally, the findings from the proposed methods and conducted experiments are utilized to answer each sub-research question in Chapter 7. This dissertation demonstrates that ground-to-aerial cross-view visual localization can become a scalable and accurate method for vehicle pose estimation. For automated driving, cross-view visual localization should not be considered a standalone task but rather a component within the vehicle localization stack. Therefore, the development of cross-view visual localization methods should account for the presence of other localization sensors. Additionally, both the accuracy and efficiency of cross-view localization can be enhanced by considering the geometric relationship between ground and aerial views. The scalability of cross-view localization methods to new areas can be improved using easily collectable noisy positioning data from the target area. Currently, although the accuracy of cross-view visual localization does not yet meet the requirements of fully autonomous driving, it can be useful for lower-level driving automation, such as Advanced Driver Assistance Systems (ADAS). Future work in cross-view visual localization should aim to further improve algorithm accuracy while considering broader uses of aerial images for automated driving.

Samenvatting

Geautomatiseerd rijden heeft een enorm potentieel om de verkeersveiligheid te verbeteren. In de afgelopen decennia is er uitgebreid onderzoek gedaan op dit gebied. Hoewel de technologische capaciteit voor sterk geautomatiseerd rijden tegenwoordig bestaat, is de grootschalige toepassing ervan nog niet aanwezig. Een belangrijke beperkende factor van de huidige oplossingen voor geautomatiseerd rijden is dat voertuiglokalisatie sterk afhankelijk is van high-definition kaarten (HD-kaarten), die zeer duur zijn om te maken en te onderhouden.

Dit proefschrift richt zich op het ontwikkelen van een meer schaalbare oplossing voor voertuiglokalisatie. Het onderzoekt een nieuwe techniek die de positie van het voertuig (locatie en oriëntatie) schat door grondniveaubeelden, vastgelegd door de onboard camera van het voertuig, te matchen met publiek beschikbare georeferentieerde luchtbeelden. Hoofdstuk 1 begint specifiek met een korte motivatie voor deze grond-tot-lucht cross-view visuele lokalisatietaak. Vervolgens definieert het de hoofd- en subonderzoeksvragen van dit proefschrift. Hierna bespreekt Hoofdstuk 2 de relevante literatuur over cross-view visuele lokalisatie. De daaropvolgende hoofdstukken behandelen cross-view visuele lokalisatie vanuit verschillende perspectieven en beantwoorden aan het eind de onderzoeksvragen.

Hoofdstuk 3 identificeert een beperking in bestaande op beeldherkenning gebaseerde cross-view lokalisatiemethoden: deze modellen zijn getraind voor grootschalige grove lokalisatie in plaats van nauwkeurige lokalisatie in geografische lokale gebieden. Deze formulering is niet geschikt voor geautomatiseerd rijden, aangezien de grove locatie van het voertuig vaak bekend is door GNSS of temporele informatie, terwijl de nauwkeurige locatie binnen een klein lokaal gebied crucialer is. Daarom stelt Hoofdstuk 3 voor om de grove lokalisatievoorkennis van andere sensoren, zoals GNSS, te integreren in de training van het beeldherkenningsmodel om het model te dwingen lokaal onderscheidende kenmerken te leren. Dit wordt bereikt door het introduceren van een nieuwe verliesfunctie, de Geo-local Triplet Loss, en de effectiviteit en algemeenheid van dit verlies wordt aangetoond met twee basismodellen op twee datasets. Bovendien wordt een veelgebruikte temporele filterpijplijn geïmplementeerd om te valideren dat het combineren van het model, getraind met het voorgestelde verlies, met GNSS-positionering betere lokalisatienauwkeurigheid oplevert dan het combineren van de basislijn met GNSS of het alleen gebruik van GNSS.

Vervolgens identificeert Hoofdstuk 4 de beperkingen van het gebruik van beeldherkenning voor nauwkeurige voertuiglokalisatie, namelijk dat betere lokalisatie meer rekenkracht en opslag vereist, omdat referentieluchtbeeldpatches dicht het doelgebied moeten bedekken. Daarom stelt het in plaats van de lokalisatie als beeldherkenningsprobleem te formuleren, voor om het grondniveaubeeld direct te matchen met een bekend luchtbeeld dat de lokale omgeving bedekt voor voertuigpositie schatting (lokalisatie plus oriënteringss schatting). Het introduceert een nieuwe methode genaamd Convolutional Cross-View Pose Estimation (CCVPE), die oriëntatiebewuste grond- en luchtbeeldbeschrijvingen construeert voor de gezamenlijke schatting van de locatie en oriëntatie van de grondniveau camera. Experi-

menten tonen aan dat CCVPE state-of-the-art nauwkeurigheid behaalt in lokalisatie- en oriënteringss schatting op drie cross-view lokalisatie benchmarks.

Daarna richt Hoofdstuk 5 zich op het verbeteren van de runtime-efficiëntie van cross-view lokalisatie. Het stelt SliceMatch voor, een generatieve testmethode die oriëntatiebewuste grond- en luchtbeeldbeschrijvingen construeert door expliciet gebruik te maken van de geometrische relatie tussen grond- en luchtbeelden. SliceMatch gebruikt vooraf berekende grond- en luchtsegmentmaskers om de functie-aggregatie te begeleiden bij het construeren van de grond- en luchtbeschrijvingen. Omdat de segmentmaskers vooraf kunnen worden berekend en de functiemaskering en descriptorvergelijking kunnen worden geïmplementeerd als matrixvermenigvuldiging, een sterk paralleliseerbaar proces, bereikt SliceMatch een snelle runtime van meer dan 167 FPS op de VIGOR-dataset.

De schaalbaarheid van cross-view lokalisatie wordt bestudeerd in Hoofdstuk 6. In de beoogde toepassing zou een cross-view lokalisatiemodel worden getraind in gebieden waar grondwaarheidsgegevens beschikbaar zijn en vervolgens worden ingezet in andere gebieden waar geen grondwaarheid is. Vanwege de domeinkloof tussen de trainings- en testregio's leidt een dergelijke directe generalisatie altijd tot een prestatieverlies. In de praktijk is het verkrijgen van nauwkeurige grondwaarheidslocatiegegevens duur en vereist het mobiele mapping-voertuigen met dure sensorkits. Het verzamelen van grondniveaubeelden met grove grondwaarheid, die fouten van tientallen meters kunnen hebben, is echter eenvoudiger, bijvoorbeeld met mobiele telefoons en hun ingebouwde GNSS. Daarom stelt Hoofdstuk 6 voor om gebruik te maken van gemakkelijk te verzamelen grove grondwaarheidsgegevens voor zwak-supervised fine-tuning van voorgetrainde cross-view lokalisatiemodellen. De grove grondwaarheidsgegevens worden gebruikt om het grondbeeld in het doelgebied te koppelen aan een luchtbeeld dat de lokale omgeving bedekt. Hoofdstuk 6 introduceert een kennis-zelfdistillatiekader dat een voorgetraind model gebruikt als leraar om pseudo-grondwaarheid te genereren voor elk grond-luchtbeeldpaar. Ruis in de pseudo-grondwaarheid wordt onderdrukt en grote uitschieters worden verwijderd met de voorgestelde technieken. Vervolgens wordt een studentmodel getraind met de verbeterde pseudo-grondwaarheid. Experimenten met twee basismodellen op twee benchmarks tonen aan dat de voorgestelde zwak-supervised kennis-zelfdistillatie tot 20% nauwkeurigheidswinst kan leiden.

Ten slotte worden de bevindingen van de voorgestelde methoden en uitgevoerde experimenten gebruikt om elke sub-onderzoeksvraag te beantwoorden in Hoofdstuk 7. Dit proefschrift toont aan dat grond-tot-lucht cross-view visuele lokalisatie een schaalbare en nauwkeurige methode kan worden voor voertuigpositie schatting. Voor geautomatiseerd rijden moet cross-view visuele lokalisatie niet als een op zichzelf staande taak worden beschouwd, maar als een onderdeel binnen de voertuiglokalisatiestack. Daarom moet bij de ontwikkeling van cross-view visuele lokalisatiemethoden rekening worden gehouden met de aanwezigheid van andere lokalisatiesensoren. Daarnaast kunnen zowel de nauwkeurigheid als de efficiëntie van cross-view lokalisatie worden verbeterd door rekening te houden met de geometrische relatie tussen grond- en luchtbeelden. De schaalbaarheid van cross-view lokalisatiemethoden naar nieuwe gebieden kan worden verbeterd met behulp van gemakkelijk te verzamelen ruwe positioneringsgegevens uit het doelgebied. Momenteel voldoet de nauwkeurigheid van cross-view visuele lokalisatie weliswaar nog niet aan de eisen van volledig autonoom rijden, maar kan het nuttig zijn voor lagere niveaus van

rijautomatisering, zoals Advanced Driver Assistance Systems (ADAS). Toekomstig werk in cross-view visuele lokalisatie zou zich moeten richten op het verder verbeteren van de nauwkeurigheid van algoritmen, terwijl bredere toepassingen van luchtbeelden voor geautomatiseerd rijden in overweging worden genomen.

1

Introduction

1

Understanding one's geographical location is fundamental for driving. For instance, navigating from a starting point to a destination requires awareness of positions. Before the wide adoption of modern automotive navigation technologies, drivers often utilized paper maps to find their global position, i.e. their location relative to an external frame of reference. This process is depicted in Figure 1.1, where a driver typically identifies his/her precise location by matching visible landmarks, such as roads and intersections, seen through the vehicle's windows, with the information presented on the map.

Figure 1.1: Humans localize themselves by comparing the surrounding environment to a paper map.

Later, the development of the Global Navigation Satellite System (GNSS), which includes the well-known Global Positioning System (GPS) and other satellite navigation systems, transferred the task of localization from the driver to the vehicle or a mobile device. GNSS provides the vehicle's global position in terms of latitude and longitude, which is then displayed on a globally registered, pre-constructed digital map. Although GNSS positioning contains errors, it generally suffices for navigation purposes for human drivers, since humans can easily determine the route using the approximate location.

However, when it comes to autonomous vehicles, the location estimate is not only used for navigation but also for driving and interacting safely with other road users, including pedestrians, cyclists, and other vehicles. In this case, GNSS alone is not sufficient. As shown in Figure 1.2, in the Oxford RobotCar dataset [1], the trajectories measured by GNSS (in red), or even more accurate Real-Time Kinematic (RTK) positioning (in green), are often noisy or miss measurements, even though the vehicle used for data collection in the Oxford RobotCar dataset is equipped with high-end GNSS receivers that consumer-level vehicles do not have. Therefore, additional localization methods that can further refine the localization accuracy on top of the noisy GNSS measurements are required.

Commonly, autonomous vehicles match measurements from their perception sensors, such as cameras and Light Detection And Ranging sensors (LiDARs), to a pre-constructed High-Definition map (HD map) for accurate localization [2–5]. HD maps contain detailed road information, such as lane boundaries, road types, and traffic lights. However, most consumer-level vehicles are not equipped with costly LiDAR sensors. Moreover, creating and maintaining an HD map is a highly laborious and expensive process [6–8]. According

to [8], HD map providers charge approximately 5000 US dollars per kilometer for mapping services in the United States. Additionally, to keep the HD maps up-to-date with urban development, frequent updates and remapping are necessary, further escalating the costs. Therefore, exploring scalable map sources for vehicle localization is crucial.

Figure 1.2: Measured trajectories in the Oxford RobotCar dataset [1]. Green: RTK measurements. Red: GNSS measurements. Note that, the selected trajectories are not labeled with "poor GPS" in the original dataset.

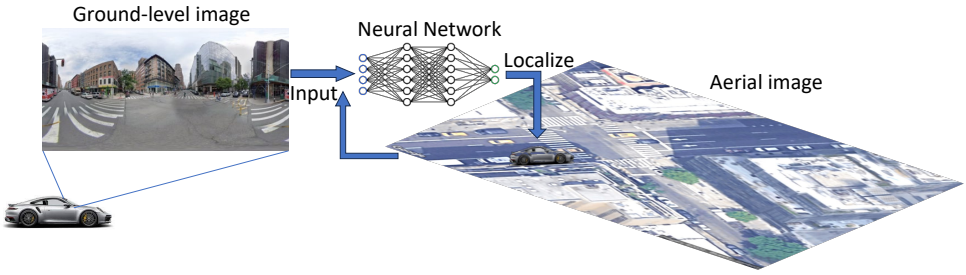


Figure 1.3: Overview of ground-to-aerial cross-view visual localization. The ground-level image captured by the vehicle's onboard camera shows that the vehicle is approaching an interaction and the surrounding four buildings have different colors and styles. The aerial image depicts the vehicle's local surroundings from a Bird's Eye View (BEV). A deep neural network then compares the information in the ground-level image to that in the aerial image to estimate the planar location and yaw orientation of the vehicle.

Aerial images contain rich information about the environment from a Bird's Eye View (BEV), making them a promising map source. Given a rough GNSS location estimate, one can retrieve an aerial image covering the local area from various sources, including web map platforms, such as Google Maps[1] and Bing Maps[2], or national government-owned geo-information web services, such as PDOK[3]. This dissertation addresses the localization of autonomous vehicles by leveraging ground-level images taken by the onboard camera and the aerial image covering the vehicle's local surroundings. As illustrated in Figure 1.3, a ground-level image, in this case, a panoramic image, shows the vehicle approaching an

[1] `https://www.google.com/maps`
[2] `https://www.bing.com/maps`
[3] `https://www.pdok.nl/`

intersection and is currently located on a zebra crossing. The surrounding buildings assist in determining on which side of the intersection the vehicle is located. A deep neural network gathers the information from the ground-level image and then compares it to the aerial image to localize the vehicle within the aerial image. This process is called ground-to-aerial cross-view visual localization. The question is then **can ground-to-aerial cross-view visual localization become a scalable and accurate method for estimating the vehicle's pose.**

Before answering this question, subsequent sections of this chapter will first highlight the motivation behind the use of autonomous vehicles, formalize the task of localization, and discuss the localization requirements for autonomous driving. Then, current techniques used for vehicle localization are introduced. Following this, the research focus of this dissertation will be presented. Finally, this chapter dissects the main question into sub-questions, outlines the remainder of this dissertation, which addresses these sub-questions, and summarizes the main contributions of this dissertation.

1.1 Why develop automated driving?

According to World Health Organization [9], in 2018, road traffic injury became the 8th leading cause of death for all age groups, and the number of annual road traffic deaths reached 1.35 million. Importantly, several leading factors of road traffic accidents are driver-related, including speeding, drunk driving, and distracted driving. These factors raise a wish: Computers shall help human drivers to increase road traffic safety. To achieve this, extensive research and engineering work has been done in automated driving, including advanced driver-assistance systems (ADAS) and autonomous driving.

The first autonomous vehicle appeared in the 1980s [10, 11]. Since then, numerous companies and research organizations have been developing and testing autonomous vehicles for various scenarios, including highway driving, dense urban driving, and challenging weather conditions. Society of Automotive Engineers (SAE) developed a classification system that defines the degree of driving automation a car and its equipment may offer [12]. From no driving automation to full driving automation, there are 6 levels:

- Level 0: no driving automation.
- Level 1: driver assistance, such as lane centering **or** adaptive cruise control.
- Level 2: partial driving automation, such as lane centering **and** adaptive cruise control.
- Level 3: conditional driving automation, i.e. the vehicle can drive by itself under limited conditions, but the driver needs to take over driving when the conditions are not met.
- Level 4: high driving automation, i.e. the vehicle can drive by itself under limited conditions, driver is not required to take over driving. For example, local driverless taxis.
- Level 5: full driving automation, i.e. the vehicle can drive by itself under all conditions.

As of 2024, several prominent companies, including Waymo, Cruise, Zoox, and Baidu, have made significant strides in deploying vehicles equipped with SAE Level 4 automation

capabilities. For example, robotaxis are operating in cities such as Seattle, Atlanta, Los Angeles, Beijing, and Wuhan.

Despite the importance of autonomous vehicles and the progress made, the market trajectory of autonomous driving has not quite matched the initial expectations of the public. Notably, the autonomous driving industry has experienced several setbacks in the past few years: Argo AI, a once-leading autonomous driving technology company supported by Ford and Volkswagen, faced bankruptcy in 2022. Ridesharing enterprises Uber and Lyft parted ways with their autonomous driving research divisions in 2020 and 2021, respectively. Automobile manufacturers, including Ford and Mercedes-Benz, have redirected their efforts from pursuing Level 4 autonomous driving to the more immediately profitable Level 2 and Level 3.

Although the technological capability for Level 4 autonomous driving exists, the widespread application and scaling of this technology present persistent challenges. One major limiting factor for its scalability is the reliance on HD Maps. HD map-based autonomous driving has been one of the most dominating autonomous driving solutions in the market, as the information possessed by HD maps largely reduces the burden for autonomous vehicles' online perception stack to sense the environment. For example, the vehicle can compare its observed landmarks to the ones labeled on the HD map to infer its location on the map. Once the ego vehicle's location is known, it also knows its relative location to other objects on the map, including the ones it cannot perceive, such as objects outside its field of view (FoV) or unobservable information like house numbers. However, creating and maintaining HD Maps on a large scale is resource-intensive, necessitating not only the use of specialized mapping vehicles equipped with costly LiDAR, high-precision GNSS, and IMU sensors but also extensive human effort in labeling data.

In general, HD maps benefit automated driving in many aspects, such as localization, trajectory prediction, and planning. This dissertation will focus on the localization of autonomous vehicles, with the goal of developing a more scalable localization solution that does not rely on HD maps. Firstly, the localization task and requirements are introduced.

1.2 Localization task and requirements

As defined in [13], mobile robot localization is the problem of determining the pose of a robot relative to a given map of the environment. Maps serve as a fixed global reference, with coordinates set independently of the robot's pose, and the process of localizing the robot in this globally registered map is also called **global localization**. The goal of global localization is then to establish correspondence between the map coordinate system and the robot's local coordinate system. Since vehicles drive on the ground, this dissertation focuses on the planar environment. Then the localization task simplifies to determining a three degrees of freedom (3-DoF) transformation between the ego-vehicle and an external planar map. The 3-DoF transformation includes the planar translation across two dimensions and its heading orientation, which is also known as yaw.

Local localization, also known as position tracking, is a sub-question in localization. Local localization focuses on estimating the motion of the robot, such as the vehicle's relative pose between consecutive timestamps. Without an already established transformation between the local motion and an external global map, the estimated trajectory of the vehicle cannot be registered globally.

To enable automated vehicles to utilize external maps, they must understand their position on the map. Thus, accurate global localization is needed, with the localization requirements varying based on factors such as road types, vehicle speeds, and vehicle dimensions.

Requirements on localization accuracy: This dissertation follows the requirements outlined in [14], where the needs for longitudinal and lateral positioning, as well as orientation estimation, are based on US road geometry standards, including lane width, curvature, etc. For passenger vehicles operating on freeway roads, the requirements are a maximum lateral error of 0.57 m, a longitudinal error of 1.40 m, and an orientation error of up to 1.50°. On local streets, due to tighter road geometry, the requirements become more stringent: lateral and longitudinal error bounds of 0.29 m and orientation accuracy of 0.50° are necessary.

Requirements on localization latency: The frequency of localization updates plays a critical role in vehicle safety. Due to sensors, algorithms, and data transfer latencies, continuous localization measurements are not feasible. At high speeds, the vehicle's location can significantly change between successive localization measurements, especially in the longitudinal direction. For instance, at 100 km/h, a 10 Hz update frequency results in localization updates every 2.8 meters, equivalent to the lane width on some local streets. At 130 km/h, 10 Hz provides updates every 3.6 meters, approximately the width of a freeway lane. According to [14], driving at 100 km/h necessitates an update rate of 150 Hz.

This dissertation aims to develop accurate and fast localization methods, with a focus on 3-DoF global localization. To avoid confusion, the remainder of this dissertation uses the term "localization" for estimating only the 2D position of the vehicle, and the term "pose estimation" is used when both the 2D position and the yaw orientation are estimated. First, the next section will discuss the commonly used sensors and techniques for vehicle localization or pose estimation as well as their shortcomings.

1.3 Current vehicle localization techniques

A variety of specialized sensors and techniques are available for vehicle localization. This section first categorizes them based on their uses for local localization or global localization. Then, the limitations of existing techniques will be discussed.

1.3.1 Local localization techniques

Local localization techniques are commonly employed in two scenarios: First, when only the vehicle's motion is needed, regardless of its global location, such as calculating the local trajectory traveled. Second, they can be combined with global localization techniques to enhance overall localization accuracy. The commonly used local localization techniques are the following:

- Inertial measurement unit: The Inertial Measurement Unit (IMU) integrates accelerometers, gyroscopes, and occasionally magnetometers to provide comprehensive motion data. It delivers three-dimensional rotation (yaw, pitch, roll) and translation information through the measurement of angular velocities and accelerations. Due

to its reliance on the double integration of these measurements over time, the IMU is prone to accumulating errors, also known as drifting. While high-precision IMUs are available and commonly used in military and aviation applications, their commercial counterparts for vehicles tend to be more affordable and less accurate.

- Wheel odometry: Vehicles employ wheel odometry to gauge the rotation of their wheels. Given that the dimensions of the wheels are predetermined, the distance traveled by each wheel can be accurately determined from their rotation measurements. While wheel odometry by itself does not measure the vehicle's exact location, the motion data it provides can be fused with other positioning sensors, such as GNSS, to enhance overall localization accuracy.

- Visual or LiDAR odometry: Visual or LiDAR odometry aims to replicate the function of wheel odometry, using only images or LiDAR scans as inputs. Typically, correspondences between image pixels or LiDAR points in consecutive measurements are established to estimate the camera's or LiDAR's relative pose at successive timestamps.

Limitations: Autonomous vehicles' perception systems are unable to capture all environmental information due to factors such as limited or blocked fields of view, sensor failure, and the inability to detect certain details like house numbers or road names. To compensate, maps are frequently employed to provide supplementary environmental information. To use these maps, the vehicle must know its global position to accurately place itself on a geo-referenced map. Local localization methods alone cannot determine this global position. Therefore, local localization techniques are typically used in combination with global localization methods to improve overall localization accuracy.

1.3.2 Global localization with satellites (GNSS)

Global Navigation Satellite System, or GNSS, is extensively utilized in various vehicle localization systems [15–17]. GNSS includes the American GPS, the Russian GLONASS, the European GALILEO, and the Chinese BeiDou. It provides geolocation and time information to a receiver anywhere on or near the Earth, when there is an unobstructed line of sight to at least four satellites.

GNSS reports the receiver's locations in the World Geodetic System 1984 (WGS84) coordinate system [18], providing longitude, latitude, and height. This information can be utilized to pinpoint the location on a globally registered reference map. In practice, various GNSS positioning solutions exist and the most prevalently applied ones are:

- The standard position service of GNSS: Accessible to all users, this service is commonly utilized by most mobile phones and consumer vehicles for GNSS positioning. It provides location accuracy ranging from several meters to tens of meters. Vertical accuracy is generally less reliable than horizontal accuracy, with precision largely depending on the number of observed satellites and environmental conditions.

- Precise Point Positioning (PPP): PPP enables high-accuracy position estimation from a single receiver without the need for proximity to a reference station. Unlike code-based standard positioning, PPP utilizes carrier-based ranging (phase measurement

of the carrier wave) to achieve location accuracy of up to 3 centimeters. However, this technique requires post-processing with precise satellite orbit and clock data. Hence it is unsuitable for real-time applications such as vehicle localization.

- Differential GNSS (DGNSS): This method can be used when there is a ground-based reference station available to broadcast the difference between the positions indicated by GNSS and the station's known fixed position. By utilizing this differential data, positioning errors can be significantly reduced, enhancing accuracy to within several centimeters.

- Real-Time Kinematic (RTK): RTK, similar to PPP, employs carrier-based ranging to achieve enhanced accuracy but also necessitates a base station with a known location, akin to DGNSS. Corrections are broadcast via radio, which requires a separate antenna to receive these signals. RTK can deliver centimeter-level accuracy [19] and is commonly utilized in mobile mapping vehicles for precise location collection, such as the reference location data in the KITTI dataset [20].

Although various GNSS positioning solutions are available, their reliability and accuracy can be compromised by several factors. The primary sources of errors in GNSS localization include [21]:

- Multipath effects: This occurs when the receiver captures GNSS signals through multiple paths, a common phenomenon in urban areas where signals are reflected off tall buildings. Areas surrounded by tall buildings are often called "urban canyons".

- Atmospheric refraction: Errors may occur due to signal refraction in the ionosphere and troposphere, which can be mitigated by modeling the atmospheric conditions or using DGNSS and RTK.

- Satellite clock drifts and orbit errors: These can lead to inaccuracies in positioning or localization, which can be corrected through the use of DGNSS and RTK techniques.

- GNSS-denied areas: In locations where satellite signals are obstructed, such as tunnels or indoor environments, GNSS cannot provide a solution. In these scenarios, localization often relies on alternative sensing technologies.

- Number of observed satellites: The accuracy of GNSS positioning depends on the number of satellites observed. Determining the location requires measurements from at least four satellites. Observing additional satellites introduces redundant measurements, thereby enhancing accuracy.

- Satellite geometry: The accuracy of positioning is influenced by the geometry of the observed satellites. If the satellites are closely clustered, resulting in small intersection angles between the lines from the receiver to the satellites, the uncertainty in planar positioning increases.

- Availability of DGNSS and RTK: The effectiveness of these high-accuracy positioning techniques depends on the presence of a GNSS base station, which may not be available in all locations.

Limitations: Even though the advanced GNSS positioning solution, RTK, can provide accurate localization for mobile mapping vehicles, consumer-level vehicles cannot use RTK due to two main reasons: first, the signal from GNSS reference stations has limited coverage; and second, the high cost prevents the inclusion of high-end GNSS receivers in consumer-level vehicles. Additionally, all GNSS positioning solutions are susceptible to the multipath effect, which is common in urban environments. Therefore, for autonomous vehicles, GNSS positioning is often complemented by other localization techniques, such as map-based localization, to ensure reliability and accuracy.

1.3.3 Global localization with maps

Map-based localization involves aligning online sensor measurements with a pre-constructed reference map. Typically, these reference maps are globally registered, which makes identifying a specific location on the map a form of global localization. For autonomous vehicles, camera images and LiDAR scans are commonly used for map matching due to their rich appearance and geometric information. The reference map can be represented in various forms. The commonly utilized ones are:

- Road maps: Road maps, such as Google Maps and OpenStreetMap, are widely used in daily life. They are primarily designed for navigation purposes that do not demand precise localization, making them less suitable for autonomous driving applications. These maps are not created with high precision and lack detailed environmental information. For instance, although roads are labeled, individual lanes are not labeled.

- Images with known pose: For large-scale coarse localization, images with known poses are also used as reference maps. During inference, a newly observed query image is compared with the reference images. The pose of the best-matched reference image is then used as the pose of the query.

- 3D point clouds: 3D point clouds accurately capture the scene's structure. When reconstructed from camera images, these point clouds can incorporate additional color information; similarly, point clouds derived from LiDAR scans may include reflectivity data. 3D point cloud maps are often collected by mobile mapping vehicles equipped with cameras, LiDAR, GNSS, and IMU. In practice, dense 3D point clouds require substantial storage, particularly for large-scale applications such as autonomous driving. Additionally, the construction of a reference point cloud map typically involves removing dynamic objects, a process that often requires human effort.

- High-definition maps (HD maps): HD maps, a concept first introduced by Mercedes-Benz in 2010 [7], aim to create a precise and informative 3D road map to support the localization, perception, and motion planning of autonomous vehicles. HD maps provide detailed representations of the environment, including roads, buildings, traffic lights, and lane markings. Typically, information in HD maps is organized into layers; for instance, the first layer may store basic road layouts, while subsequent layers contain more detailed information like lane markings and attributes. Different companies, such as TomTom and HERE, have developed different map layer structures [3]. The industry also developed a variety of HD map data formats and standards, e.g.

1

OpenDRIVE[4] and Navigation Data Standard (NDS) Open Lane Model[5]. OpenDRIVE defines road using a reference line and offsets from the reference line, while NDS Open Lane Model uses lane center line and boundaries. Conceptually, HD maps can be viewed as a lane graph with detailed attributes, including connectivity, lane types, and driving directions. A common method for constructing HD maps involves extracting information from multi-sensor measurements, such as camera images and LiDAR point clouds [7]. Alternatively, road networks and lane information can also be derived from aerial images [7, 22, 23]. Given the high localization accuracy requirement for autonomous driving, HD map-based localization has become the standard solution.

Limitations: Although HD map-based localization is accurate, its scalability is limited by the costs associated with constructing and maintaining an HD map. Firstly, the data collection requires operating a fleet of mobile mapping vehicles equipped with expensive sensors. Furthermore, labeling road lanes and traffic components demands human effort, which becomes costly in practice for large mapped areas. These factors make the HD map-based solution challenging to scale to less developed areas or rapidly developing regions, which require frequent map updates. In practice, a more scalable solution for vehicle localization is required.

1.3.4 Simultaneous localization and mapping

Simultaneous Localization and Mapping (SLAM) [24–27] can be used for both local and global localization. It is a technique that constructs or updates a map of an unknown environment while simultaneously keeping track of the robot's location within the constructed map. SLAM typically utilizes camera images or LiDAR scans, with the map being represented as a point cloud. In use cases where the global location of the robot is not crucial, such as with vacuum cleaning robots, SLAM is used solely for local localization. For vehicle localization, the constructed map often needs to be globally registered, and then SLAM integrates also global positioning estimates such as those from GNSS.

Usually, SLAM consists of a front end and a back end. The front end focuses on estimating motion, similar to visual or LiDAR odometry. This results in a graph of poses. The back end concentrates on optimizing and registering this pose graph. It requires loop closure, i.e. revisiting the same location multiple times to generate redundant measurements for error correction, and global localization.

Limitations: In autonomous driving, SLAM is typically utilized in constructing the reference map [28], but it is less common for the online operation of autonomous vehicles due to its high computational cost and storage requirements. Commonly, map construction relies on data collected by specialized mapping vehicles equipped with high-end sensors. Consumer-level vehicles outfitted with standard sensor kits may not achieve the required accuracy for mapping.

[4] https://www.asam.net/standards/detail/opendrive/
[5] https://nds-association.org/

1.4 Cross-view localization, an alternative?

Ground-to-aerial visual cross-view localization, or cross-view localization for short, aims to localize the ground-level camera by matching the image it captures with a georeferenced aerial image. Hence, cross-view localization is a form of map-based global localization. In practice, the size of the aerial image varies depending on different use cases. When there is no prior knowledge of the ground-level camera's location, the aerial image needs to cover the entire Earth. However, for vehicle localization, a rough estimate of the vehicle's location is often available, resulting in a smaller reference aerial image size. Besides, since the relative pose between the vehicle center and the onboard camera is fixed, the estimated camera pose can be converted into the vehicle pose. Usually, the output of cross-view localization is the 2D planar location of the ground camera in the aerial image, along with its yaw orientation, see Figure 1.3.

One major advantage of cross-view localization lies in the widespread availability of aerial images.

Aerial images: Aerial images are images captured from an airborne platform, including fixed-wing aircraft, helicopters, unmanned aerial vehicles (UAVs or "drones"), etc. Aerial images taken at an angle are called oblique aerial images, and those taken straight down are vertical aerial images. Vertical aerial images are often used for landscape monitoring and mapping, such as in photogrammetry and cartography. High-resolution aerial images can have spatial resolution of up to 1-5 centimeters per pixel. Nowadays, relatively low-resolution (around 10 centimeters per pixel) aerial images are publicly available from various sources, including web map platforms, such as Google Maps[1] and Bing Maps[2], or government-owned geo-information web services, such as PDOK[3]. Depending on the sources, the update frequency of their aerial images differs, for example, PDOK updates its aerial images every year, while Google Maps updates its aerial images every 1-3 years. This dissertation will use those publicly available aerial images.

Because of perspective projection, vertical aerial images still capture a small part of building facades and the scale is not uniform inside the image, i.e. the ground distance of each pixel is not equaled. Orthographic rectification corrects it and turns aerial images into aerial maps, that have uniform scale. However, since a digital elevation model (DEM) is required to create an accurate aerial map, aerial maps are not widely available. In practice, the scale difference in the aerial image is small, and hence this dissertation will directly use aerial images as the reference map.

Potential benefits: Compared to HD maps, which are expensive to construct, aerial images already provide global coverage, making them a more scalable map source. Additionally, the information in HD maps is extracted by humans. For instance, lane boundaries and traffic signs are crucial for humans to perform driving tasks, and hence these objects are extracted and labeled. However, these manually extracted features might be sub-optimal for training a deep neural network for localization. On the other hand, aerial images are direct projections of the 3D scene. Despite lacking explicit human annotations, aerial images contain more information about the environment and might enable better localization.

Cross-view localization and GNSS positioning are complementary techniques. Compared to GNSS-based localization, which often suffers from large errors in urban areas due

to the multipath effect, urban environments offer rich visual cues for matching ground-level and aerial images, such as buildings, intersections, and lane markings. Consequently, cross-view localization may achieve higher accuracy in urban areas compared to highways, where surroundings often feature repetitive patterns, and GNSS tends to be more reliable due to the open sky. On the other hand, consumer-grade vehicles are mostly equipped with GNSS receivers. Although GNSS may have a positioning error of tens of meters, it still provides a rough localization estimate that narrows down the search area in the aerial image. This dissertation will assume the existence of such a rough GNSS localization prior.

Challenges: Cross-view localization involves matching the ground-level image with the aerial image. The drastic differences in viewpoint and scale between ground-level and aerial images introduce challenges to this matching process. Additionally, since ground-level and aerial images are not collected simultaneously, rather, aerial images are captured at an earlier time, only static objects visible in both views are matchable, while dynamic objects are not. Hence, the achievable accuracy of cross-view localization remains unclear. Moreover, the learned features of cross-view localization models should be generalizable across different regions, despite potential differences in scenes between training and test regions. Despite the strong capability of deep learning models in learning to extract relevant features from data, only a limited amount of literature on cross-view localization has studied the aforementioned challenges, as it is a relatively new field of research. This dissertation aims to bridge this gap.

1.5 Research questions and chapter outline

First, this section will define the main research question of this dissertation and then break it down into sub-questions. Subsequently, an outline of the following chapters will be presented.

1.5.1 Research questions

Given the challenges involved in matching ground-level and aerial images, the main research question of this dissertation is defined as:

MQ: Can ground-to-aerial cross-view visual localization become a scalable and accurate method for estimating a vehicle's pose by comparing its captured ground-level image with an aerial image (the "map") covering its local surroundings?

To address this main research question, this dissertation will develop a deep neural network for ground-to-aerial cross-view visual localization. Previously, cross-view localization has primarily been approached by matching ground-level and aerial images using image retrieval methods. Therefore, the first emerging sub-question is:

SQ1: Is the common image retrieval formulation in ground-to-aerial cross-view image matching well-suited for vehicle localization?

Once a suitable formulation for cross-view image matching is identified, the focus will shift to estimating the vehicle's pose (location plus orientation). Consequently, the next sub-question is:

SQ2: How can the location and orientation of the ground-level camera be jointly estimated?

Additionally, as outlined in the localization requirements in Section 1.2, localization latency is an important factor in autonomous driving. Thus, the next sub-question focuses on the efficiency aspect of cross-view localization methods, namely:

SQ3: What strategies can be employed to create an efficient ground-to-aerial cross-view visual localization method?

After addressing the previous sub-questions, this dissertation will delve into one of the two key aspects of the main research question: the scalability of cross-view localization methods. The focus here is the methods' scalability to new regions. Firstly, this dissertation will investigate whether ground-to-aerial cross-view visual localization methods can generalize to new regions without being trained on any data collected in the target regions. Given that modern deep learning methods often experience a performance drop when directly generalizing to distributions different from the distribution of training data, this dissertation will also study how to mitigate this performance drop caused by the domain gap. Then, the next sub-research question is:

SQ4: Do ground-to-aerial cross-view visual localization methods generalize to new regions, and how can their scalability be enhanced with easily collectable data?

Following the previous sub-question, the final sub-question focuses on the other key aspect of the main research question: the pose estimation accuracy of cross-view localization.

SQ5: What level of accuracy is achievable with ground-to-aerial cross-view visual localization?

1.5.2 Chapter outline

To answer the research questions of this dissertation, the following chapters are structured as follows. First, Chapter 2 on related work categorizes visual localization methods and reviews relevant approaches for ground-to-aerial cross-view localization. Following this, the Chapter 3 to 6 address the sub-questions.

Chapter 3 follows the common image retrieval formulation for ground-to-aerial cross-view image matching-based localization. Additionally, it also considers the rough localization prior from GNSS. It proposes a method where the local region of a continuous aerial image is densely sampled into patches and matched with ground-level images for localization. Besides, this chapter develops a temporal filtering pipeline to fuse cross-view image retrieval and GNSS positioning over time. Chapter 4 follows the idea of considering the GNSS localization prior but formulates cross-view localization differently to address the computational and storage demands of image retrieval for precise localization. It develops a method that directly correlates the representation of a ground-level image with that of a local aerial image patch to jointly estimate the vehicle's location and orientation. Chapter 5 focuses on the efficiency aspect of cross-view localization and develops a novel generative-testing cross-view localization approach. Chapter 6 addresses the scalability problem. Given the high cost of obtaining accurate ground truth data for training cross-view localization models in a supervised manner, Chapter 6 proposes a weakly supervised learning approach to scale the cross-view localization methods to new areas without accurate ground truth. The effectiveness of the proposed framework is demonstrated using the algorithm developed in Chapter 4.

1

Finally, Chapter 7 concludes the dissertation by summarizing key findings and answering the research questions. It reflects on the progress made, highlights remaining challenges, and discusses the derived insights. Furthermore, it suggests directions for future research to bridge the gap between current cross-view localization techniques and the demanding localization requirements of autonomous vehicles.

1.6 Contributions

The core achievement of this dissertation lies in advancing ground-to-aerial cross-view image matching for vehicle localization. The key contributions are outlined as follows:

Integration of noisy localization priors in cross-view image retrieval: The rough location of the vehicle can be estimated by many means, such as specialized sensors and temporal filtering. Typically, errors in such location estimates can reach tens of meters, for example, GNSS positioning in urban canyons. Previously, cross-view image retrieval localization was often treated as a substitute for GNSS for global rough localization, but it has limited localization accuracy in smaller local areas. Chapter 3 argues that, instead of training cross-view image retrieval models as an alternative to GNSS, rough location estimates should be incorporated into the training of these models to enhance accuracy in locally ambiguous areas.

Specifically, a novel Geo-local Triplet Loss is proposed to enforce cross-view image retrieval models to learn image representations that are specifically discriminative between images from geographically nearby locations, rather than for distant areas, which was the main objective in previous works. To test generalization across recording days, Chapter 3 also augments the well-known Oxford RobotCar dataset with a map composed of aerial image patches to serve as a new dense cross-view image retrieval benchmark. The real-world utility of the proposed method is demonstrated through a scenario where cross-view image retrieval localization is fused with actual GPS measurements in a particle filter pipeline. Chapter 3 shows that the proposed method significantly enhances localization accuracy and robustness compared to the baselines.

The content in Chapter 3 is published in the European Conference on Computer Vision 2020 Workshop on Map-based Localization for Autonomous Driving [29] and in IEEE Robotics and Automation Letters, 2021 [30].

Joint fine-grained localization and orientation estimation: Formulating localization as a retrieval problem introduces a trade-off between the localization accuracy and the density of the reference aerial patches sampled from the target area. Therefore, Chapter 4 moves one step further by addressing the task of fine-grained cross-view localization, i.e. identifying the precise location of the ground image inside a known aerial image that covers the local surroundings. This chapter proposes a novel method, Convolutional Cross-View Pose Estimation (CCVPE), for this task.

CCVPE surpasses the previous state-of-the-art baselines by a large margin in localization and achieves comparable orientation estimation accuracy on VIGOR and KITTI datasets. It constructs a multi-modal distribution for localization and uniquely associates each location with its most probable orientation. It avoids a dense search over all 3-DoF poses

(localization plus orientation) by discretizing the orientation sparsely and performing additional regression. This formulation is efficient for fine-grained pose estimation. It is also shown that the predicted probability can be used to filter out predictions that potentially have large localization or orientation errors. The proposed architecture exploits the strength of a translational equivariant feature encoder and contrastive learning. Its ground image encoder maintains the spatial scene layout information relative to the camera's viewing direction in the ground image descriptor and the contrastive loss enforces aerial descriptors to encode global orientation information. These descriptors enable joint localization and orientation estimation with negligible extra computational cost.

The content in Chapter 4 is published in the European Conference on Computer Vision 2022 [31] and in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024 [32].

Enhancing the efficiency of fine-grained cross-view localization: Previous state-of-the-art fine-grained cross-view localization methods have slow inference time. To improve the runtime efficiency of fine-grained cross-view localization, Chapter 5 proposes a generative-testing approach, SliceMatch. It has a novel aerial feature aggregation step that uses a cross-view attention module for ground-view guided aerial feature selection, and the geometric relationship between the ground camera's viewing frustum and the aerial image to construct pose-dependent aerial descriptors. SliceMatch's design allows for efficient implementation, which runs significantly faster than previous state-of-the-art methods. Namely, for an input ground-aerial image pair, SliceMatch extracts dense features only once, aggregates aerial descriptors at a set of poses without extra computation, and compares the aerial descriptor of each pose with the ground descriptor.

The content in Chapter 5 is published in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023 [33].

Scaling cross-view localization models to new areas without ground truth: Fine-grained cross-view localization methods typically rely on accurate ground truth data to train a deep neural network. In practice, acquiring accurate ground truth is a laborious and expensive process. To tackle this, Chapter 6 proposes a knowledge self-distillation-based weakly-supervised learning approach that considerably improves models' localization performance in a new area by only leveraging the ground-aerial image pairs without ground truth locations. For methods with coarse-to-fine outputs, this chapter investigates how to reduce the uncertainty and suppress the noise in the teacher model's predictions. Using the proposed single-modal pseudo ground truth leads to a better student model than using the multi-modal heat maps from the teacher model. This chapter also designs a simple but effective method for filtering outliers in the pseudo ground truth. Training with filtered pseudo ground truth further improves the localization accuracy of the student model.

This content in Chapter 6 is published in the European Conference on Computer Vision 2024 [34].

2

Related work

2.1 Visual localization

Visual localization is the problem of estimating the pose of a camera relative to a reference scene representation, based on an image captured by the camera. This field has been explored through various methodologies, broadly categorized into absolute pose regression, image retrieval-based localization, structure-based localization, relative pose estimation, and foundation models, see an overview in Figure 2.1. The task of ground-to-aerial cross-view localization has been approached through both image retrieval and relative pose estimation methods. This section introduces the basics of different formulations, and the subsequent section will provide a detailed review of the relevant literature in cross-view localization.

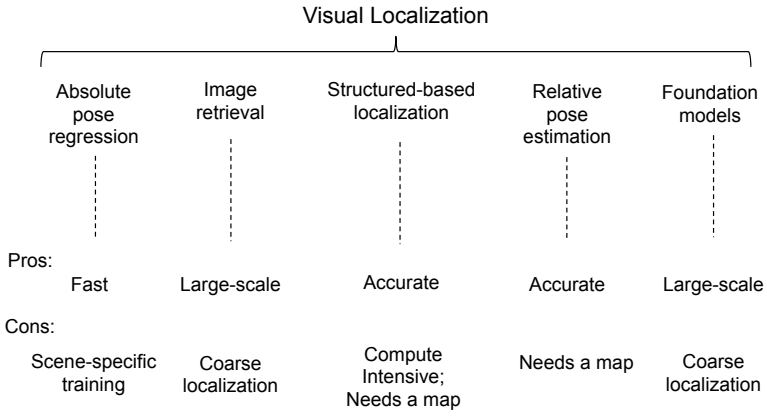


Figure 2.1: Overview of visual localization categories and their pros and cons.

2.1.1 Absolute pose regression

Absolute pose regression refers to directly regressing a pose of a given query image relative to a pre-defined coordinate frame.

PoseNet [35] pioneered this research by taking a single RGB image as input and outputting the 6 DoF camera pose of the input image. During training, the network implicitly encodes the external reference coordinate frame of the training images into its learned parameters. The predicted pose is defined on this external reference coordinate frame. Subsequent research has studied developing more complicated network architectures [36, 37], advanced loss functions [38], gathering temporal information [39], as well as jointly estimating auxiliary tasks [40–43].

Absolute pose regression shares common insights with recent advances in Neural Radiance Fields (NeRF) [44], as both rely on the model parameters to represent the scene. The main difference is that NeRF explicitly leverages the projection geometry and maps coordinates to the appearance of 3D points, while absolute pose regression focuses on mapping on the image level and maps image appearance to camera pose.

Compared to other visual localization techniques, absolute pose regression methods usually have fast runtime, since only one network forward pass on a single image is

required. In recent developments, absolute pose regression is also used for vehicle localization [45, 46]. However, the main downside of absolute pose regression is its generalization capability across different areas [47]. The necessity for the model to encode the scene and its coordinates means that deploying the model in a new location demands retraining specific to that area.

2.1.2 Image retrieval-based localization

Image retrieval-based localization aims to identify the most similar image in a reference database to a given query image, and then uses the camera pose of the most closely matched reference image as the pose of the query image. The similarity between the query and reference images is measured based on their image descriptors.

Before the advent of deep learning, these descriptors were typically composed of aggregated hand-crafted features, such as Bag-of-Visual-Words [48, 49], Vector of Locally Aggregated Descriptors (VLAD) [50–52] and Fisher vector [53, 54].

With the introduction of NetVLAD [55], the focus shifted towards learning-based approaches, which have proven superior in extracting and identifying relevant features due to their ability to adapt and generalize across variations in viewpoint, lighting, weather conditions, and the presence of dynamic objects.

Learning-based methods branch into two directions: one [55–58] focuses on extracting holistic features from the whole image. A deep network is used to embed the full image into a single image descriptor, usually a 1-D vector, without explicitly enforcing the locality of features. Those descriptors can be more robust against dynamic objects since the model might learn to ignore those objects by reasoning from the full image content. Another branch [59–65] tries to learn representative local features, such as landmarks and key points, as they can be more robust against changes in viewpoints.

Since the query image pose is approximated by the pose of the retrieved reference image or an interpolation of poses of several top retrieved images, the density of the reference images directly influences the localization accuracy. Therefore, image retrieval-based localization is often used for large-scale coarse localization to provide an initial pose estimate that is then refined by a more accurate localization method.

2.1.3 Structure-based localization

Structure-based localization methods match the query image to the structure of the scene to estimate the camera pose using projection geometry.

Typically, the scene is represented by images with poses or 3D point clouds. Then, solving structure-based localization can be done by registering the query image into a local Structure-from-Motion (SfM) pipeline [66] or establishing query image pixel-to-3D point matches and solving the Perspective-n-Points problem inside RANSAC [67]. The key to structure-based localization comes down to an accurate feature-matching method that establishes correspondences between pixels across images, or pixels to 3D points.

Establishing pixel correspondences involves two phases: detection and description. In [68], SIFT [69] is used for key points detection, and then semantic information is embedded into the learned descriptor. SuperPoint [70] and D2-Net [71] use a CNN to detect key points and generate for them feature descriptor simultaneously. SuperGlue [72] trains a graph neural network to match the local features from two images. LoFTR [73] makes

use of the self and cross-attention layers in Transformer to obtain feature descriptors from two images and perform the matching.

For matching pixels to 3D points, [74] converts the extracted features from the query image into visual vocabularies and directly matches them to the pre-constructed visual vocabularies of 3D point clouds. In [74], a fast 2D-to-3D matching scheme is developed for fast structure-based localization.

While local feature matching stands as a fundamental and continually evolving field of research, this summary does not encompass the full breadth of existing methodologies. Empowered by advanced local feature matching technologies, structure-based localization is capable of achieving centimeter-level accuracy in position estimation and sub-degree precision in orientation. However, these methods are often constrained by their computational intensity and the necessity for pre-existing 3D models or densely captured reference images with accurate poses. The difficulty of capturing and updating the accurate 3D model at a large scale as well as the storage overhead limits the application of structure-based localization methods to large-scale applications, such as autonomous driving.

2.1.4 Relative pose estimation

Relative pose estimation techniques focus on determining the positional and orientational relationship between a query image and a reference map. The form of the reference map is diverse. Structure-based localization can be seen as a form of relative pose estimation with the 3D structure of the scene used as the map. One can also use an image with a known pose as the reference map [75, 76]. Ground-to-aerial cross-view localization can also be formulated as relative pose estimation, with the aerial image as the reference map. Since this topic will be covered later, this section mainly discusses relative pose estimation methods using other BEV maps than aerial images.

OpenStreetMap contains nodes, edges, and polygons with semantic labels describing the topological structure of the environment including buildings, road networks, etc. It has been used as a map source for vehicle localization with both camera [77–79] and LiDAR sensors [80, 81]. Recent advance [77] maps the camera image into a BEV representation and compares it to the deep features extracted from the map at all possible 3 DoF poses.

HD maps are commonly utilized for both visual [82, 83] and LiDAR-based [84, 85] localization. Similar to visual localization based on OpenStreetMap, localization with HD maps also involves comparing objects observed in camera images to those annotated in the HD map. Given that HD maps are constructed with high precision and contain a range of semantic labels, localization based on HD maps can obtain high accuracy.

Floor map-based localization tries to localize the camera inside a building's BEV floor map of walls and rooms. LaLaLoc [86] renders ground-view floor layouts from a BEV floor map and learns a shared descriptor space for query images and rendered layouts for end-to-end retrieval and pose refinement. LaLaLoc++ [87] removes the need for the rendering step in LaLaLoc [86] and uses a UNet-like architecture [88] to build a descriptor at each candidate location. Localization is achieved by looking for locations whose local map descriptor is similar to the descriptor of the query. Laser [89] renders ground descriptors from a floor plan in an efficient way and formulates localization as metric learning.

The accuracy of relative pose estimation depends greatly on the form of the maps. Maps with detailed geometry and semantic information about the environment, such as

HD maps, can enable high localization accuracy. However, constructing these maps can be expensive and laborious.

2.1.5 Foundation models

Foundation models [90–94] have demonstrated remarkable capabilities across a broad spectrum of language and vision tasks. Recently, foundation models for image geo-localization [95–97] have also emerged. They follow the formulation of PlaNet [98] and CPlaNet [99] by dividing the World into cells and classifying the images across "geographical cells". So far, the focus of geo-localization foundation models is to identify the rough location of the image over the world and have limited localization accuracy, e.g. a few kilometers, making them less relevant for vehicle localization. Still, as evidenced in other vision tasks [92], with the increase in model size and amount of data, those models can potentially be useful for vehicle localization.

2.2 Ground-to-aerial cross-view localization

Before the widespread application of deep learning, ground-to-aerial cross-view localization was addressed by detecting hand-crafted local features, such as SIFT features [69], or local image patches like building facades [100], from both views and matching them [101]. However, the large perspective difference between ground-level and aerial views makes hand-crafted features perform poorly.

With deep learning, the domain gap between ground-level and aerial views can be minimized in the learned feature space. Depending on the use case, cross-view localization has been addressed by either image retrieval or relative pose estimation. The subsequent subsections will discuss these approaches individually.

2.2.1 Cross-view image retrieval

Cross-view image retrieval has shown great progress in the past years. It enjoys the advantage of the widely available geo-referenced aerial images and aims for rough geo-localization by retrieving the aerial image patch that covers the location of the ground-level query image. The first deep networks for this task date back to 2015 [102–104]. Since then, the common practice of using Siamese-like architecture was established. The ground and aerial images are encoded into image descriptors by two network branches. Usually, these branches do not share weights [105–107], because the two input images are from different domains. This domain gap is also one of the main challenges in the cross-view setting. Subsequent works seek to bridge the domain gap between the learned ground and aerial representations via various approaches.

An effective way for minimizing the domain gap is to construct visually similar inputs [106, 108–111]. SAFA [106] observes that the polar rays in the aerial image correspond to the vertical lines in the ground image, and proposes to use a polar transformation on the aerial image to build an image that is visually similar to the ground view. In [111], an inverse polar transformation is used on ground-level panoramas to generate synthetic aerial images. In [108], the authors bridge the domain gap between the ground and aerial images by generating synthetic aerial images using GANs [112]. In [109], ground-level images are generated from aerial images using GANs, and the features for image generation

are shared for cross-view image retrieval.

Besides constructing visually similar inputs, several works try to optimize the learned image feature for retrieval in different ways. CVM-Net [105] adopts the powerful image descriptor, NetVLAD [55], to learn how to gather local image features for building global image descriptors. In [107], the authors propose to use the orientation information of both views to guide the model to find more discriminative features across views. CVFT [113] considers Optimal Transport theory to facilitate the feature alignment between ground and aerial images. Global-assists-local [114] addresses the case of retrieving a ground-level query with a limited horizontal FoV and proposes to embed the aerial feature outside the query's FoV into the aerial descriptor to aid the retrieval. In [115], the feature locality is explicitly enforced when building global image descriptors by partitioning the encoded features. CVLNet [116] gathers temporal information into the ground descriptor by making use of a ground-level query video. Recently, transformers are also used. L2LTR [117] introduces self-cross attention to flow effective information into the descriptors. TransGeo [118] proposes an attention-guided non-uniform cropping method to attend to and zoom in the informative local image patches. Apart from retrieval, several works also estimate the orientation of the ground camera [119–122].

However, the major limitation of cross-view image retrieval is that the ground query is assumed to be located at the center of the matched aerial image patch. In practice, it is not possible to have aerial image patches centered at unknown test locations. Densifying reference aerial patches reduces the influence of this assumption but increases the computation cost. In [110], the authors propose to zoom into the initial retrieved aerial image and crop smaller aerial patches at a set of candidate locations in the initial retrieved image for second-stage retrieval. A few works [123–125] fuse the image retrieval results with temporal filters for more accurate localization. Still, estimating an accurate location and orientation of a single ground-level query image within a reference aerial image patch remains an open yet important task.

A recent survey [126] on image geo-localization provides an in-depth analysis of cross-view image retrieval-based localization on their methodologies and performance. For readers seeking detailed insights in this area, this dissertation recommends referring to the findings and discussions presented in the survey [126].

2.2.2 Cross-view camera pose estimation

Cross-view camera pose estimation can be seen as a follow-up task after image retrieval or other coarse localization techniques. Given a ground-level query and an aerial image that covers the local surroundings of the query, the objective is to estimate the exact location and the orientation of the query within the given aerial image. In [127], a large-scale dataset for this task is introduced, and the authors propose a model that first retrieves an aerial image given the ground query with a known orientation and then regresses the location offset between them. Later, [128] also formulates the localization as a regression problem and includes an additional road extraction training objective. In [129], the orientation of the ground camera is estimated by assuming the location of the ground camera in the aerial image is known. Instead of regression, [130] solves the query ground camera pose by iterative optimization. It first warps the feature from the aerial image to a ground view using a homography and then uses a multi-level Levenberg-Marquardt algorithm

to estimate the 3-DoF ground camera pose using the warped aerial feature and extracted ground-level feature. In [131], a Recurrent Homography Estimation module is used for estimating the relative pose between the projected ground-level feature and aerial feature. SNAP [132] fuses the information from ground-level and aerial images to construct a neural map for localizing other ground-level images. Vision Transformers [133] are used in [134] to map the features of the ground-level surrounding views to BEV, and the mapped BEV feature maps are densely compared to feature maps extracted from the aerial image for pose estimation. In [135], LiDAR measurements are fused with camera images for cross-view pose estimation.

However, there are several limitations in the above methods. Some methods only estimate the location [127, 128] or orientation [129] of the ground camera. Current regression [127, 128] or optimization [130, 135] formulation for localization restricts the output to a single mode without uncertainty estimation. When there are several visually similar locations in the aerial view, regression-based methods [127, 128] might regress to the midpoint between those locations, and optimization-based methods [130] might converge to a wrong local optimum. More importantly, these methods lack uncertainty estimation to reflect the quality of the outputs. Besides, the runtime is also a bottleneck in many existing methods, e.g. 2 to 3 FPS in [130, 134, 135].

2.3 Localize other modalities on aerial images

Range sensing sensors-to-aerial image localization received a lot of attention. RSL-Net [136] localizes Radar scans on a known aerial image patch. This task is formulated as generating a top-down synthetic Radar scan conditioned on the aerial image using [137] and then comparing the online scan to the generated synthetic scan for pose estimation. Later, this idea is extended to self-supervised learning [138]. In [139], the top-down representation of a LiDAR scan is compared to UNet [88] encoded aerial features for localization. The range information is crucial in representing the measurement in a top-down view and thus making the measurement comparable to aerial images.

Despite aerial images can be used to localize different modalities, this dissertation will focus on using ground-level images.

3

Cross-view image retrieval for vehicle localization by learning geographically local representations

This chapter is based on *Z. Xia, O. Booij, M. Manfredi, and J. F. P. Kooij, "Geographically Local Representation Learning with a Spatial Prior for Visual Localization," European Conference on Computer Vision Workshops, pp. 557-573, 2020* [29], and *Z. Xia, O. Booij, M. Manfredi, and J. F. P. Kooij, "Cross-View Matching for Vehicle Localization by Learning Geographically Local Representations," IEEE Robotics and Automation Letters, vol. 6, no.3, pp.5921-5928, 2021* [30].

3.1 Overview

With the rise of camera-equipped vehicles, visual localization has become a key research topic in autonomous driving. No matter how the map is presented, most visual localization methods explicitly or implicitly match an input image to a representation of the map. For instance, image retrieval-based methods localize the query image by matching it to the geo-referenced images in a shared representation space. An increasingly popular variant is cross-view image retrieval-based localization [102, 103, 105–107, 119], where the query ground-view image is compared to aerial or satellite imagery. This setting enjoys the reliable representation and dense coverage of the environment from the overhead view. Plus, large databases are nowadays readily available [103, 107].

In the robotics domain, localization is traditionally addressed using specialized sensors, e.g. Global Navigation Satellite Systems (GNSS). Unfortunately, the horizontal positioning error of stand-alone GNSS can reach tens of meters [140, 141] near high-rising buildings or under trees, due to the multipath effect. In practice, the GNSS localization is often fused with measurements from other sensors, e.g. wheel odometry or camera, and combined with temporal filtering.

All recent cross-view image retrieval-based localization methods [105–107, 113, 121] target large-scale global localization and have demonstrated decent performance [123] on a mobile mapping vehicle. However, substantial gaps still exist in how the localization task is addressed in mobile robotics and autonomous driving, and the state-of-the-art image retrieval-based localization techniques.

First, image retrieval-based localization is often treated as a substitute for GNSS for global place recognition [123, 142], though in practice GNSS and temporal filtering can provide a complementary coarse location estimate [143]. Second, existing cross-view image retrieval benchmarks [107, 120] measure how the model generalizes to new areas, as they split the data according to its geographic region. However, in practice, aerial images of the test region are often available during training, especially for a navigation task with geo-localized road information, which already presupposes that the target region is known. Therefore, an equally relevant question is how the learned representation generalizes to new ground-level observations on different days in the same area. Third, many cross-view image retrieval-based localization methods [106, 107, 113, 121] are evaluated solely using metrics designed for retrieval, such as recall@K. Such metrics do not measure the actual localization capability, and do not reflect that a ground image's view does not necessarily correspond to any reference aerial image patch's center location, or could even coincide with multiple overlapping aerial image patches.

To address the observed gaps, this dissertation exploits the context of cross-view image retrieval within a localization system. Since other components, e.g. GNSS and temporal filtering, will already provide a coarse location estimate, this dissertation proposes to train cross-view image retrieval to be especially discriminative within this local region of uncertainty, rather than differentiating far-away areas that the prior would already discard, see Figure 3.1.

The main contributions of this chapter include: (i) A novel Geo-local Triplet Loss that enforces cross-view image retrieval models to learn image representations that are specifically discriminative between images from geographically nearby locations, rather than for distant areas. The effectiveness of the proposed loss function is validated with

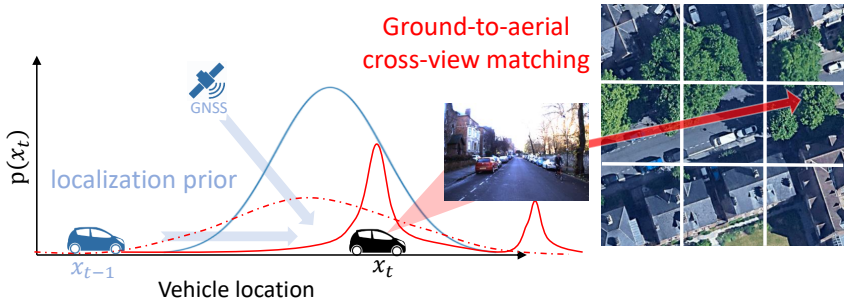


Figure 3.1: Vehicles can use cross-view image retrieval between camera images and aerial image patches for self-localization, resulting in a geo-global localization estimate (red dashed curve). However, a coarse localization prior (blue curve) is often already available from other sensors or temporal integration. This prior can be exploited during training to obtain a more discriminative model within the local area (red solid curve).

two state-of-the-art methods (ii) This chapter augments the well-known Oxford RobotCar dataset with a map composed of aerial image patches to serve as a new dense cross-view image retrieval benchmark to test generalization across recording days. Experiments are also conducted on data from the existing CVACT benchmark, for which this chapter proposes new splits, to test generalization across regions. On both benchmarks, quantitative improvements over the state-of-the-art are demonstrated. Qualitatively, the difference between encoded geo-local and geo-global features can be observed. (iii) The proposed approach is tested in a real-world scenario where query images are matched against aerial image patches distributed evenly in the target area, and the cross-view image retrieval-based localization is fused in a particle filter with priors from actual GPS measurements. This chapter demonstrates superior localization accuracy and robustness against the baseline cross-view image retrieval fused with GPS.

3.2 Methodology

This section starts by reviewing the task of cross-view image retrieval and the triplet loss used in the baseline and related work. After this, the proposed geo-local loss is explained. Finally, a particle filter is introduced for combining cross-view image retrieval scores and GNSS measurements for online vehicle localization.

3.2.1 Cross-view image retrieval task

Given a *ground-level* query image G_q, the objective of cross-view image retrieval is to select the closest *aerial* image patch from the target dataset $\mathbb{A} = (A_1, A_2, \cdots)$. Each aerial image patch A_i here covers a fixed-sized square area of the Earth's surface, and the 2D geographic location $\xi(A_i) \in \mathbb{R}^2$ of the center of the square is known. The retrieval is done by matching images in a representation space, where the aerial images and query are mapped into normalized image descriptors using mapping function $f(\cdot)$ and $g(\cdot)$ respectively. The descriptor of the best-matched aerial image should have the smallest squared Euclidean distance to the descriptor of the query.

3.2.2 Baseline architecture and geo-global triplet loss

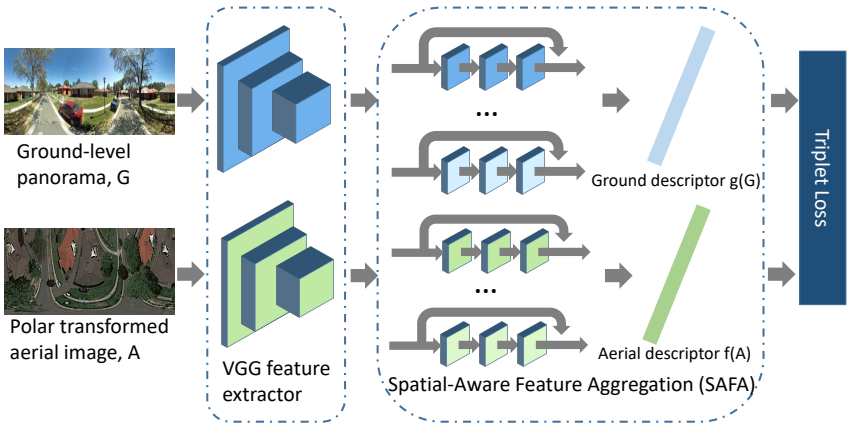


Figure 3.2: The architecture of the baseline cross-view image retrieval method, SAFA [106].

While the proposed approach is generic, the state-of-the-art SAFA method [106] is used as the main baseline. As shown in Figure 3.2, the mapping functions $f(\cdot)$ and $g(\cdot)$ in SAFA are implemented as a 16-layer VGG feature extractor and 8 separate spatial-aware feature aggregation modules [106]. They map input images to 4096-dimensional descriptors. Two network branches without weight-sharing are trained on image pairs $\mathbb{X}_{\text{train}} = \{(A_1, G_1), (A_2, G_2), \cdots\}$ using a soft-margin triplet loss for two related matching objectives,

$$\mathcal{L}_1(i,j) = \log(1 + e^{\gamma(d_{i,i}^2 - d_{i,j}^2)}), \qquad \text{(aerial-to-ground)} \tag{3.1}$$

$$\mathcal{L}_2(i,j) = \log(1 + e^{\gamma(d_{i,i}^2 - d_{j,i}^2)}). \qquad \text{(ground-to-aerial)} \tag{3.2}$$

Here $d_{i,j} = \|f(A_i) - g(G_j)\|_2$ is the Euclidean distance between the descriptors, and γ is a hyperparameter to adjust the gradient of the loss. The final loss is the average of $\mathcal{L}_1(i,j)$ and $\mathcal{L}_2(i,j)$. For a minibatch $\mathbb{B} \subseteq \mathbb{X}_{\text{train}}$ of B pairs, the loss terms can be efficiently computed by performing the forward passes $f(A_i)$ and $g(G_i)$ only once for all B samples, and then just computing B^2 Euclidean distances $d_{i,j}$ of all combinations i, j.

An important aspect of the baseline is that it selects minibatches from the training data by randomly shuffling *all* samples in each epoch, thus any two pairs are equally likely to co-occur in the batches, independently from their geographic proximity. This triplet loss thus learns a *globally* discriminative representation.

3.2.3 Training with a geo-local triplet loss

Vehicle localization provides at every time step a coarse localization estimate from fusing and filtering past sensor measurements. This chapter therefore seeks to exploit knowledge of a coarse prior already during training, and will consider two adaptations to the baseline loss, namely *geo-distance weighted loss terms* and *local minibatches* [29].

Geo-distance weighted loss terms This chapter proposes to multiply the triplet losses in Equation (3.1) and (3.2) with a weight $w_{geo}(i,j)$ that scales their contribution based on the Euclidean distance in meters $\delta_{i,j} = \|\xi(S_i) - \xi(S_j)\|_2$ between their geographic positions $\xi(S_i)$ and $\xi(S_j)$,

$$w_{geo}(i,j) = p_r(\delta_{i,j}) \cdot (1 - e^{-{\delta_{i,j}}^2/(2\sigma^2_{\text{geo}})}). \tag{3.3}$$

The first term $p_r(\delta_{i,j})$ models a prior on the coarse localization error, which is assumed to be maximally r meters. Importantly, it should force training to ignore triplets with $\delta_{i,j} > r$ and in favor of nearby ones. Two options are considered for p_r. Option 1 uses a *step* function to weigh all triplets 1 if $\delta_{i,j} \leq r$, and 0 otherwise [29], see the green dotted line in Figure 3.3. Option 2 uses instead a *Gaussian* function with a standard deviation $r/3$ such that the weight smoothly drops to (nearly) zero at r meters, see red dotted line in Figure 3.3. The second term is added to down-weight the loss on geographically *nearby* samples to prevent the model from treating two nearly identical aerial images, e.g. with a distance of 1 meter, one as positive and the other one as negative. The hyperparameter σ_{geo} controls the smoothness of this weight reduction.

The full weight function $w_{geo}(i,j)$ is thus the product of both terms, and scaled such that the weight at its maximum is 1, see the green/red solid lines in Figure 3.3 for the final weight function with a step/Gaussian decay.

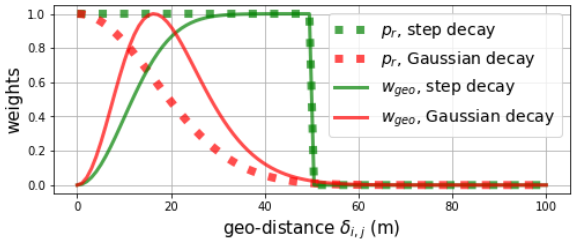


Figure 3.3: The weight decay options (dashed) and resulting weight functions $w_{geo}(i,j)$ (solid), here shown as an example of $r = 50$m and $\sigma_{\text{geo}} = 10$m.

Local minibatches Using the geo-distance weighted loss term, most randomly picked pairs from the training data would have zero weight as they are likely to be at distant geographic locations, especially when the mapped area is large. This chapter therefore proposes to construct local minibatches that only contain pairs from nearby geographic locations, using the following procedure:

1. pre-compute before training for each pair $P_i = (S_i, G_i)$ the local neighborhood of pairs within a geographic radius of r meters, i.e.

$$\mathbb{N}_r(i) = \{(S_j, G_j) \mid i \neq j \wedge \delta_{i,j} \leq r\} \subset \mathbb{X}_{\text{train}}. \tag{3.4}$$

2. At the start of an epoch, create a fresh set $\tilde{\mathbb{X}}$ containing all training samples, $\tilde{\mathbb{X}} \leftarrow \mathbb{X}_{\text{train}}$, representing the still unused samples in this epoch.

3. To create a new minibatch $\mathbb{B}$ of size B, first randomly pick a pair P_i from pool $\tilde{\mathbb{X}}$, and then uniformly pick without replacement the remaining $B-1$ samples from the neighborhood set $\mathbb{N}_r(i)$. All picked samples are removed from the epoch's pool, $\tilde{\mathbb{X}} \leftarrow \tilde{\mathbb{X}}/\mathbb{B}$. Once $\tilde{\mathbb{X}}$ is empty, a new epoch is started.

Since all pairs j in the batch are by definition within distance r from the first sampled pair i, two samples j and j' in the minibatch can be *at most* a distance of $2r$ meters apart. This local minibatch formulation greatly increases the chance that many pairs in the minibatch are also within each other's r-meter radius, and thus largely reduces the chance of near-zero geo-distance weighted loss terms. Note that overall each pair occurs in *at most* one minibatch per epoch. Pairs without enough neighbors will not be used.

Importantly, r is a measure of the coarse prior's maximum uncertainty, thus it is not an optimizable hyperparameter but is given by the targeted localization use case. To avoid minibatches with too few samples, the selected training data should contain at least $B-1$ neighbors within a radius of r of each sample.

3.2.4 Particle filter-based localization

This section then describes how online vehicle localization could use cross-view image retrieval at test time, and fuse it with real-world GNSS measurements in a temporal filter, as opposed to replacing the GNSS (e.g. [123, 142]). The distribution of the localization results, which will be multi-modal, is captured by constructing a particle filter-based localization pipeline [13] that combines the cross-view image retrieval and GPS positioning. It is assumed that aerial images $\mathbb{A}_{grid}$, which cover the target area centered around points on a dense regular grid, are available.

Each particle m has a 4D state vector $x^{[m]}$ containing Easting, Northing, forward velocity, and yaw in the map's coordinate frame. Let χ_t denote the set of $M = 2000$ particles at step t. At $t = 0$, all particles are initialized at the GPS-measured location with random yaw between $-180°$ and $180°$ and a random velocity between 0 and $5\text{m}/s$. For $t > 0$, a prediction is made for each particle χ_{t-1} using a fixed velocity motion model with Gaussian acceleration and steering noise. The particles are then weighted by the measurement model, and finally resampled proportional to weight to obtain χ_t. As filter output, the median of each element in the state vector over all particles in χ_t is taken.

The measurement model weighs each particle ${x_t}^{[m]}$ according to the query image $G_{q,t}$ and the raw GNSS positioning $\xi(G_{q,t})$. Assuming that the GPS uncertainty follows a Gaussian distribution with known standard deviation σ_{gps}, a confidence threshold of this distribution is set to $3\sigma_{\text{gps}}$. Then the aerial image $\mathbb{A}_{local} \subseteq \mathbb{A}_{grid}$ within the threshold, i.e. a 2D circle centered at $\xi(G_{q,t})$ with a radius of $3\sigma_{\text{gps}}$ meters, are select from the database. Particles outside this circle are directly discarded, and only aerial images $A_j \in \mathbb{A}_{local}$ are compared to $G_{q,t}$ to compute their cross-view image matching score $e^{-d^2_{j,q}}$. Given $\xi(m)$, the Easting and Northing location of ${x_t}^{[m]}$, let $e^{-d^2_{m,q}}$ be the geo-distance-based bi-linear interpolation of the matching scores for the 4 aerial images at the grid points around $\xi(m)$. The particle's weight is then,

$$w_t[m] = \frac{e^{-d_{m,q}^2}}{\sum_{j \in \mathbb{A}_{local}} e^{-d_{j,q}^2}} \cdot e^{-\|\xi(m)-\xi(G_{q,t})\|_2^2/(2{\sigma_{\text{gps}}}^2)}. \tag{3.5}$$

Here the first term computes the probability of the query being located at $\xi(m)$ as given by cross-view image retrieval. This probability equals the matching score at $\xi(m)$ over the sum of scores between the query and all aerial images in $\mathbb{A}_{local}$. The second term in the equation measures the likelihood of $\xi(m)$ being the correct location according to the raw GPS measurement. Equation (3.5) thus presents a straightforward sensor fusion of the visual cross-view retrieval and GNSS localization measurements, and is applicable irrespective if the matching network is trained geo-local or geo-global.

Unfortunately, GPS measurements inevitably carry huge errors in extreme cases, for example when no satellites are in sight. Motivated by the outlier rejection found in [13, 144], such situations are handled by not using the GPS-measured location at step t when this is over $r_{outlier}$ meters apart from the GPS measured location at step $t-1$ or if there is no valid GPS measurement at this step. Instead of the raw GPS, the estimated location at step $t-1$ as $\xi(G_{q,t})$ is then used. The $r_{outlier}$ is set to $3 \cdot \sigma_{\text{gps}} + v_{t-1} \cdot \Delta t$, where v_{t-1} is the estimated velocity at previous step and Δt is the time interval.

3.3 Experiments

The proposed geo-local representation learning is compared to the standard geo-global representation learning [106] in two scenarios, namely, generalization across regions and generalization across time. Besides quantitative results on two retrieval benchmarks, this section also provides a qualitative view of the uncertainty of the localization and extracted features. Lastly, this section will show that the benefits of the proposed cross-view matching approach on the retrieval benchmarks also translates to a realistic localization task using the particle filter and real GPS measurement data.

3.3.1 Datasets

Here, two image retrieval benchmarks are discussed. While ideally the training data is collected according to the application-specific r (see Section 3.2.3), to reuse existing datasets in the following experiments, a suitable target r value is assumed by considering each dataset's sample density.

CVACT Dataset: CVACT [107] is a large cross-view dataset with GPS footprint for image retrieval. It contains 35532 ground panorama and aerial image pairs, denoted as CVACT_train, and 92802 pairs as CVACT_test. Notably, the validation set CVACT_val of 8884 pairs is a subset of CVACT_test, and [106] reported their quantitative results on the CVACT_val rather than CVACT_test. This chapter will not follow the data split in [107],[106], because CVACT_val is rather sparse, and it trivializes the task formulation of localization using a prior too much as it discarded all negative samples. Furthermore, this chapter follows the target use case where all aerial images are available during training and split only the ground images into training, validation, and test set.

Oxford RobotCar Dataset: Oxford RobotCar [1],[145] is a dataset targeted at autonomous driving and contains images, raw GPS recordings, RTK measurements, etc.,

under different lighting and weather conditions collected in different times of the day and over a year in multiple traversals in the Oxford region.

The dataset has not been used for cross-view image matching-based localization, as it does not contain aerial images. To construct a novel benchmark, this chapter collected aerial images at zoom level 20 (~ 0.0924 m/pixel) with the Google Maps Static API for each ground-level front-camera image. The aerial images were cropped into 600 pixel × 600 pixel, which corresponds to a 55.44 m × 55.44 m ground area, and the ground-level images are cropped to exclude visible parts of the ego-vehicle.

(a) (b) (c)

Figure 3.4: Three sample pairs in the proposed Oxford RobotCar cross-view localization benchmark highlight some local and global differences. (a) and (b) are 5 m apart, (b) and (c) are 20 m apart. Ground images are from different traversals and recording days, resulting in variations in cars, vegetation, and lighting conditions.

This chapter does not target the most extreme lighting and weather conditions and hence selects the traversals recorded at different daytime and days with the label "sun", "overcast" or "clouds" and which contain both raw GPS and accurate RTK localization measurements. In the dataset, the front-viewing images are taken at 16 Hz. To make sure the consecutive ground images do not look too similar in appearance, the images are sub-sampled to guarantee that there is at least 5 m between two consecutive frames in each traversal. Finally, the corresponding aerial images centered at the ground truth locations are collected to build the ground-to-aerial pairs. In total, there are obtain 23854 pairs from 15 traversals. The training (17067), validation (1698), and test (5089) sets contain ground images from 11, 1, and 3 traversals. All aerial images are used during training, validation, and testing. The chosen test traversals are collected in Summer (Test 1,2) and Winter (Test 3) with labels "overcast, roadworks" (Test 1), "sun" (Test 2), and "overcast" (Test 3) to include variations in season, weather, and road conditions from the training set. For the same season and weather conditions, the test traversals are collected at a later time

of day than the training recordings. In this dense dataset, almost all images have more than 200 pairs in a $r = 50$ m neighborhood. Some example ground and aerial pairs are shown in Figure 3.4.

In addition, this chapter also collected aerial images to cover the Oxford region at a grid with 5 m interval, similar to [123]. This data will be used as the database $\mathbb{A}_{grid}$ for the particle filter of Section 3.2.4 to simulate a real-world localization task with the dataset's raw GPS and front-camera video stream.

3.3.2 Network architecture and implementation details

The baseline SAFA method [106] is implemented based on the code released by its authors. To implement the proposed method, the same network architecture is used. Only the loss is replaced by the proposed geo-distance weighted loss, and the model is trained using local minibatches[1]. Both the baseline model and the proposed model are trained on introduced data splits following the same procedure as in [106]: The VGG part is pre-trained on Imagenet [146], Adam [147] is used as optimizer with a learning rate of 1×10^{-5} on the CVACT dataset and 5×10^{-5} on the Oxford RobotCar dataset. In the triplet losses, $\gamma = 10$, and the dropout keep rate is set to 0.8. On the traversal-based split Oxford RobotCar dataset, an additional dropblock [148] with a block size of 11 and keep probability of 0.8 is used to reduce overfitting.

On the CVACT dataset where the ground images are 360° panoramic views, polar transformed aerial images are used, as done in [106]. Due to its sparseness, the only correct match for a query ground image is the aerial image patch centered the exact same location. On the dense Oxford RobotCar dataset, it is observed that defining the training objective as matching the query to the aerial image at the exact location is too strict and the validation loss struggles to decrease. Therefore for a query ground image this chapter selects a random aerial image at a small geospatial offset of a maximum of 5 m, which is the same distance used to subsample camera frames (see Chapter 3.3.1) As additional data augmentation, the aerial image patches are also rotated by a random multiple of 90°.

3.3.3 Evaluation metrics

This chapter will consider two aspects in evaluation, namely image retrieval performance on the benchmarks, and localization performance for the particle filter.

For the retrieval task, this chapter assumes at test time a known (worst-case) prior localization error of radius r, and thus directly discards for both the baseline method and the proposed method any false negatives beyond r meters of the true location. Still, for reference, the case when no such prior would be available (i.e. an *infinite* test radius) is also reported. The recall@1 and recall@x meters are the quantitative metrics. They measure how often the top-1 retrieved aerial image is located at the exact location of, or less than x meters away from, the ground truth location. Although a maximal 5 m geospatial offset is introduced in selecting matched aerial images for each query during training, the recall with $x < 5$ m will still be reported to give an overview of how top-1 retrieved aerial images distribute during testing.

[1]The data (with an overview of time, season, label of chosen traversals) and code is available at `https://github.com/tudelft-iv/Visual-Localization-with-Spatial-Prior`

As motivated in Section 3.1, recall does not fully reflect a model's localization performance. In the particle filter experiments, this chapter instead measures the Euclidean localization error in meters between the true location and the median particle location during each traversal, and report the mean, median, 90%-quantile, 95%-quantile, and 99%-quantile error.

3.3.4 Effect of key hyperparameters

In this section, the impact of three key hyperparameters: batch size N, weight decay d_r, and smoothness σ_{geo} are tested.

This chapter experimented with batch sizes $N = 4, 16, 64$. The batch size directly influences the training stability as it defines how many negative pairs are used when one positive pair is presented. On the Oxford RobotCar dataset [1],[145], the training "collapsed" (i.e. descriptors are filled with only zeros) with small batches $N = 4$ / 16, around epoch 4 / 421 for our model, and around epoch 20 / 712 for the baseline. It is found that this behavior is due to values in the image descriptor (before normalization) exceeding numerical limits. Adding extra regularization does not prevent this. However, when $N = 64$, those values are kept under a much smaller magnitude. It is reckoned by this chapter that when the batch size is too small with limited diversity in the aerial images, there is a risk that only maximizing the similarity between positive pairs is enough to push negative samples away in representation space, and the network will put very large weights on such similarities. Indeed, on the sparse but diverse CVACT dataset training does not collapse with $N = 4$ or $N = 16$ for either model. Unfortunately for $N = 64$ many locations will not have sufficient neighbors to fill the batch. This chapter therefore keeps $N = 16$ for CVACT, and $N = 64$ for Oxford RobotCar. In general, since r upper-bounds N, a small r requires dense training data to avoid potential training instability.

To choose between step decay and Gaussian decay for p_r, other hyperparameters are kept the same and the model is trained with different decay options for the proposed loss on the Oxford RobotCar dataset. The model trained with step decay surpasses the model trained with Gaussian decay by a large margin of 16.9% in recall@5m. It is notable that, for the same r, the Gaussian decay heavily down-weights far away samples, however, these contribute greatly to the validation performance. This chapter will therefore use $w_{geo}(i, j)$ with the step decay in the later experiments.

This chapter tests $\sigma_{\text{geo}} = 0,\ 5,\ 10,\ 15$ meters, and finds validation recall@5m is 75.5, 79.9, 82.5, 81.2 percent respectively on the Oxford RobotCar dataset ($\sigma_{\text{geo}} = 0$m indicates no down-weighting of nearby negative samples). Clearly, in this dataset where images are densely distributed, down-weighting the nearby negatives samples is important to learn a good representation. On the sparse CVACT dataset, it is observed that σ_{geo} does not influence performance much. For the remainder, $\sigma_{\text{geo}} = 10$m is used for both datasets.

3.3.5 Generalization across regions

The experiment on the CVACT dataset shows how well the learned representation generalizes to unseen ground images in new areas. Since locations are more sparsely distributed, here $r = 100$m is used as a weak hypothetical localization prior to train the model. To test the generality of geo-local training, this chapter directly applies it to another baseline method, DSM [121], in addition to the regular SAFA baseline without further geo-local loss

hyperparameter-tuning.

All models are trained for 100 epochs, and the best ones are kept according to validation split performance.

It is observed that geo-local models converge faster than the baselines even though the geo-distance weighted loss assigns zero weights to some triplets in the local minibatches. Evaluation results are reported on the test split in Table 3.1. Providing the same localization prior to testing, models trained with the proposed loss improved the recall@1 by around 12.3% (74.0 vs 65.9 percent) for SAFA, and around 3.2% (70.4 vs 68.2 percent) for DSM. Meanwhile, they also beat baselines by a considerable margin in terms of the recall@5m and recall@10m. Importantly, these results confirm that the proposed geo-local representation does not capture features that *identify* the local training region, which would not generalize, but captures features that *discriminate* nearby locations, which does generalize. Furthermore, the improvements of geo-local method generalize over different baselines, without the need for any baseline-specific hyperparameter tuning. As expected, globally (i.e. with ∞ test radius) the models trained with the proposed loss perform worse than the baselines, as it violates the prior assumption. Still, in real-world applications, a coarse localization estimate is often available to benefit from geo-local features.

Table 3.1: Evaluation on CVACT Test Set (best in bold). The term "local" means the model trained with the proposed geo-local loss.

Recall@	1	1	5 m	10 m
Test Radius	100 m	∞	100 m	100 m
SAFA[106] (%)	65.9	**59.9**	68.8	78.2
SAFA-local (%)	**74.0**	55.8	**77.5**	**85.4**
DSM[121] (%)	68.2	**64.0**	71.5	80.4
DSM-local (%)	**70.4**	56.6	**73.9**	**82.0**

3.3.6 Generalization across time

On the Oxford RobotCar dataset, this chapter tests how well the learned representation generalizes to new ground images collected on other dates and at different times of the day in the same region. Since the images are distributed much denser here, a more realistic hypothetical localization prior, $r = 50$ m, is used. The best model is kept according to the validation performance in 1000 epochs of training.

The quantitative test results of the selected model are summarized in Table 3.2. When the localization prior is available, the geo-local representation learned by training with the proposed loss consistently outperforms the baseline on all test traversals. Overall, the proposed approach generalizes well across time-of-day and different days, and it does not overfit on the training ground images or specific time and weather conditions, which is important as in practice localization in the target region will be done on different days.

3.3.7 Qualitative results

This section illustrates how the proposed model performs differently from the baseline. To provide a qualitative view of the model behavior, the localization heat map is visualized using the similarity measurement between a test query and all nearby aerial images. On

Table 3.2: Evaluation on Oxford RobotCar Test Sets (best results in bold). T.N stands for the N-th testing traversal, and the mean recall over 3 traversals is in the bottom row.

Recall@	1	1 m	3 m	5 m	5 m
Test Radius	50 m	50 m	50 m	50 m	∞
T.1 baseline[106] (%)	7.1	26.3	75.5	92.3	**90.6**
T.1 proposed model (%)	**9.7**	**38.4**	**84.5**	**96.0**	64.3
T.2 baseline[106] (%)	5.3	19.5	59.4	81.9	**76.1**
T.2 proposed model (%)	**8.3**	**29.6**	**71.8**	**85.9**	43.0
T.3 baseline[106] (%)	5.7	21.4	62.0	83.4	**79.5**
T.3 proposed model (%)	**8.4**	**28.9**	**77.2**	**88.9**	53.0
Mean baseline[106] (%)	6.0	22.4	65.6	85.9	**82.1**
Mean proposed model (%)	**8.8**	**32.3**	**77.8**	**90.3**	53.4

both CVACT, Figure 3.5a, and Oxford RobotCar, Figure 3.5b, the proposed model outputs a sharper localization result inside the prior area, while the baseline has more uncertainty about the exact location along the road. Unlike the baseline, the proposed model also produces other high-probability peaks outside the circle. This is because it does not distinguish distinct areas, and similar local spatial layouts may reoccur elsewhere. This trade-off comes from the geographically local representation the proposed model uses.

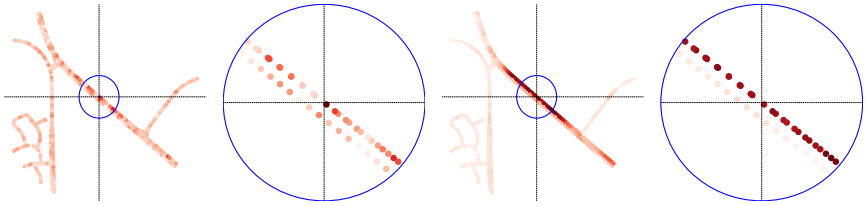
This is verified by comparing the encoded image features of both approaches. Similar to [106], the spatial embedding maps are back-propagated to the input image to show where the model extracts features [149], see Figure 3.6. On the CVACT dataset, the proposed model pays attention to vegetation and streetlights. The baseline model, on the other hand, ignores these objects and focuses on the road structure. On the Oxford RobotCar dataset, the proposed model looks for traffic lights and building facades, while the baseline mostly looks at the canopies and building roofs. The objects the proposed model pays attention to are repeated at many different places, nevertheless, they are useful in disambiguating other images along this road. The baseline focuses on fewer environmental details, which are sufficiently discriminative globally but not locally.

3.3.8 Temporal filtering

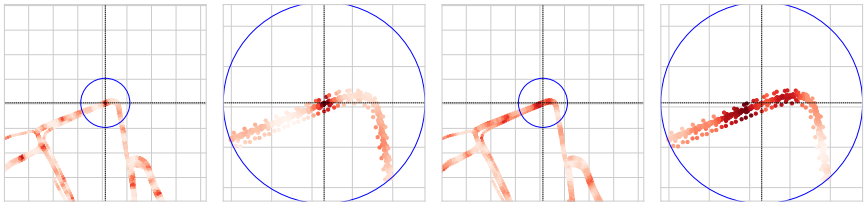
Finally, this section validates that the better performance of the proposed model in the discussed benchmarks also translates to actual gains in a real-world localization task using actual GPS measurements and temporal filtering priors, as opposed to hypothetical priors. The localization pipeline is tested on the Oxford RobotCar dataset with an update rate of 1.6 Hz, where every 10th image from the unsampled test traversals is used as the ground-level query and is matched to regularly distributed aerial images. No additional sensors are included in the temporal filter, such as wheel odometry and IMU, to keep the amount of tuneable system configurations and parameters to a minimum.

The quality of the GPS measurements controls the hyperparameter σ_{gps}. Unfortunately, the GPS error is often unpredictable and can vary significantly. For example, the mean error of the raw GPS positioning on Oxford RobotCar test traversals is around 3.7 m, but reaches 13 m on the validation traversal. In conducted experiments, the σ_{gps} is set to 10 m.

The quantitative results over 3 test traversals are summarized in Table 3.3. Temporal



(a) Zoomed out/in views of localization heat map on CVACT dataset (left: proposed model, right: baseline [106])



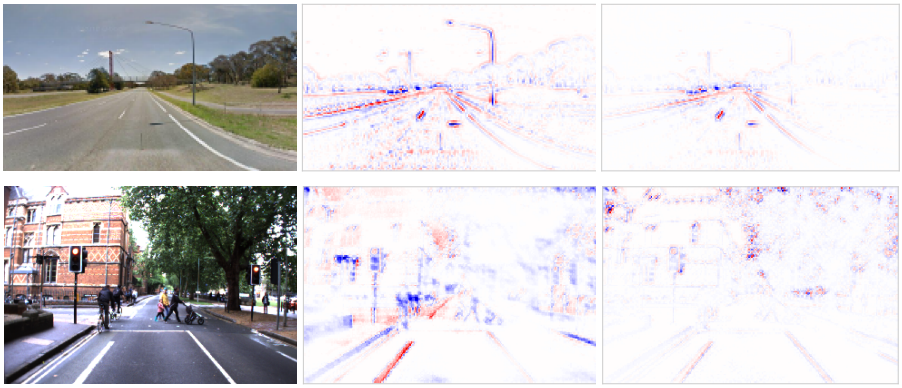
(b) Zoomed out/in views of localization heat map on Oxford RobotCar dataset (left: proposed model, right: baseline [106])

Figure 3.5: Examples of localization heat maps on (a) CVACT and (b) Oxford RobotCar dataset. Each dot represents an aerial image with the darkness proportional to the similarity to the query image. The ground truth location of the query is indicated by the cross. The circle indicates the local neighborhood with radius $r = 100$m in (a) and $r = 50$m in (b). The zoomed-out image shows the surrounding 1km × 1km area in (a) and 400m × 400m area in (b). On both datasets, the proposed approach results in a single peak within the local neighborhood, while the baseline has more uncertainty.

filtering of raw GPS alone in the particle filter achieves an average error of ~ 5 m. Incorporating the cross-view matching improves localization significantly, especially when GPS produces spurious large outliers, as seen from the 99%-quantile error. An example is shown in Figure 3.7. Importantly, the proposed model delivers overall the best accuracy and robustness, and reduces mean (2.77 m vs 3.32 m) by 17% and 99%-quantile error (9.97 m vs 13.83 m) by 28% compared to the baseline.

The superiority of the proposed method together with GPS and particle filter comes from the sharp cross-view matching result. Most of the time, using GPS is enough for global coarse localization, and adding another coarse estimate from global cross-view matching does not gain much in localization accuracy. In contrast, the proposed method effectively refines the GPS positioning within GPS-uncertain areas. Extreme erroneous GPS measurements are filtered out by the outlier rejection module in the temporal filter, ensuring a reasonable prior is obtained from previous time instances.

Note that, in other regions where there are many high-rising buildings, a larger σ_{gps} could give better localization results. However, it is observed on the validation traversal that also for different σ_{gps} values in the range from 5 m to 30 m the proposed model still outperforms the baseline, and does not influence the conclusion here.

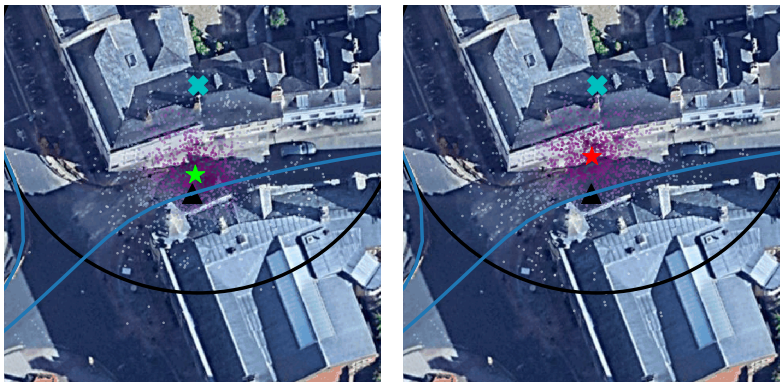


(a) Input (b) Proposed model (c) Baseline

Figure 3.6: Visualized back-propagated encoded feature attention maps for a ground image in CVACT dataset (first row) and Oxford RobotCar dataset (second row).

Table 3.3: Particle filter localization error (mean and error at x%-quantile) on Oxford RobotCar test traversals. Best results in bold. "baseline+GPS" and "proposed model+GPS" use both the cross-view matching module and GPS. "GPS" is without any cross-view matching.

Localization error (m)	mean	50%	90%	95%	99%
T.1 GPS	4.66	3.93	8.24	10.73	20.89
T.1 baseline[106]+GPS	3.23	2.63	5.71	7.27	14.91
T.1 proposed model+GPS	**2.65**	**2.12**	**4.70**	**5.91**	**11.00**
T.2 GPS	4.50	4.00	7.48	9.28	19.19
T.2 baseline[106]+GPS	3.19	2.71	5.58	7.02	11.90
T.2 proposed model+GPS	**2.73**	**2.46**	**4.71**	**5.73**	**8.12**
T.3 GPS	4.64	3.92	8.76	10.64	20.51
T.3 baseline[106]+GPS	3.53	2.72	6.69	8.30	14.69
T.3 proposed model+GPS	**2.94**	**2.49**	**5.46**	**6.92**	**10.80**
Mean GPS	4.60	3.95	8.16	10.22	20.20
Mean baseline[106]+GPS	3.32	2.69	5.99	7.53	13.83
Mean proposed model+GPS	**2.77**	**2.36**	**4.96**	**6.19**	**9.97**



(a) Localization result (proposed) (b) Localization result (baseline [106])

Figure 3.7: Particle filter-based localization. Each purple dot (particle) has a darkness (re-sampling weight). The cyan cross and black circles show the raw GPS positioning and its 95% confidence interval. The black triangle marks the ground truth location on the full trajectory (blue line). The green (or red) star is the localization result of using the proposed model (or the baseline) in the pipeline.

3.4 Conclusion of the chapter

In this chapter, the cross-view image retrieval method is embedded into a workable real-world localization system by considering the prior from other localization components in the training. It was quantitatively and qualitatively showed that the advantage of geo-local training over geo-global training on state-of-the-art methods. A 12.3% improvement of the recall@1 was achieved on the CVACT dataset. On the Oxford RobotCar dataset, the recall@5m was improved from 85.9% to 90.3%. Besides, it was also demonstrated that the increase in cross-view matching capability translates to 17% lower mean and 28% lower 99%-quantile localization error when real GPS measurements and cross-view matching scores are fused in a particle filter-based localization pipeline. More importantly, all noticeable quantitative benefits come from a simple-to-implement and generic adaptation.

4

Convolutional cross-view pose estimation

This chapter is based on *Z. Xia, O. Booij, M. Manfredi, and J. F. P. Kooij, "Visual cross-view metric localization with dense uncertainty estimates," European Conference on Computer Vision, pp. 90-106, 2022* [31], and *Z. Xia, O. Booij, and J. F. P. Kooij, "Convolutional Cross-View Pose Estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 5, pp. 3813-3831, 2024* [32].

4.1 Overview

Cross-view image matching has shown notable performance in large-scale geo-localization [102, 103, 105–107, 119] by formulating the problem as image retrieval. However, the assumption that query ground images correspond to the center of aerial patches in the database does not hold during test time. Besides, formulating the problem as a retrieval problem introduces a trade-off between the localization accuracy and the density of the aerial patches of the target area. In practice, global localization can also be obtained by other means in outdoor robotics, such as temporal filtering or coarse GPS/GNSS [29, 30, 136], but can still have errors of tens of meters [29, 30, 140]. This chapter follows [29, 30, 136] by exploiting a coarse location estimate, and zooms into fine-grained camera pose estimation within a known aerial image, i.e. to identify which image coordinates in the aerial patch correspond to the location of the ground camera and the orientation of the ground camera.

However, several gaps must be filled before large-scale real-world deployment of cross-view camera pose estimation methods is a realistic possibility for self-driving. So far, the localization accuracy of existing methods is not yet good enough for autonomous driving requirements, e.g. the lateral and longitudinal error should be below 0.29 m [14]. Besides, many methods cannot be run at sufficiently low latency, i.e. ~15 frames per second (FPS), on datasets for self-driving[1, 20, 150]. For example, [130] relies on iterative optimization to estimate the ground camera's pose. In [134], computationally heavy Transformers are used to construct Birds Eye View (BEV) feature representations, and then the BEV representations from ground and aerial views are compared densely at each of the location-orientation combinations (i.e. 3-DoF poses). Both methods [130, 134] run at a low frame rate, e.g. 2 to 3 FPS. It is also observed that when the aerial view contains a symmetric scene layout, e.g. at crossroads, single-mode regression-based methods [127, 128] might regress to a midpoint between visually similar locations, and optimization-based methods [130] might get stuck at a wrong local optimum.

To improve the pose estimation accuracy over prior works and meanwhile achieve fast runtime, This dissertation proposes a novel method that predicts a multi-modal distribution for localization and jointly considers the orientation of the ground camera. As shown in Figure 4.1, the translational equivariance property of convolutional networks is exploited to construct orientation-aware image descriptors that represent visual information in both ground and aerial views at different locations with a particular viewing direction. Joint localization and orientation estimation are achieved by convolving the ground descriptor on the aerial descriptors with circular padding, i.e. matching the ground descriptor to different rolled/shifted versions of the aerial descriptor. Then, the proposed model regresses the fine-grained orientation based on discrete orientation matching scores and follows a coarse-to-fine formulation to gradually refine a sparse location map into dense output. The final output orientation is conditioned on the predicted location.

The main contributions of this chapter are: (i) This chapter proposes a novel method for end-to-end cross-view camera pose estimation, Convolutional Cross-View Pose Estimation (CCVPE)[1]. It surpasses the previous state-of-the-art baselines by a large margin in localization and achieves comparable orientation estimation accuracy on VIGOR and

[1]Code is available at `https://github.com/tudelft-iv/CCVPE`

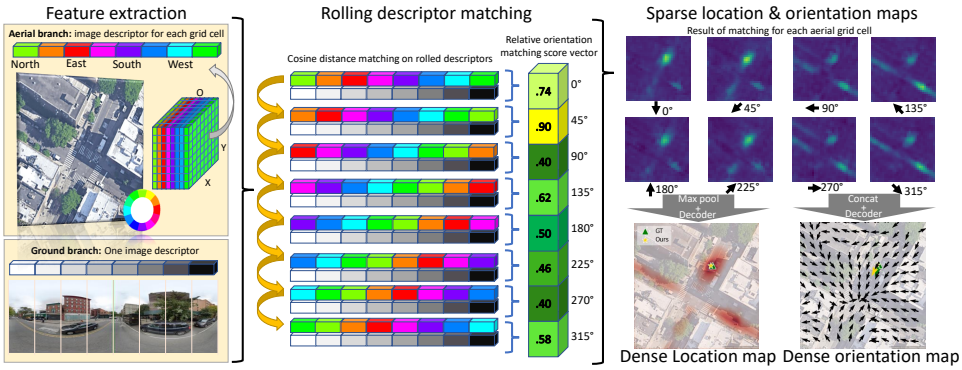


Figure 4.1: In Convolutional Cross-View Pose Estimation (CCVPE), ground and aerial images are encoded into orientation-aware image descriptors. For the aerial image, a grid of descriptors are created. Efficient joint localization and orientation prediction are enabled by matching rolled aerial descriptors with the ground descriptor. Sparse location and orientation maps are up-sampled into dense maps using decoders with coarse-to-fine matching. To predict the most probable location considering different orientations, the descriptor matching scores are max-pooled over orientation channels. The matching scores from different orientations are concatenated to gather information for accurate orientation prediction. The final orientation prediction is conditioned on localization, i.e. it is selected at the predicted location in the dense orientation map.

KITTI datasets when testing generalization to new measurements within the same area and across different areas. (ii) CCVPE constructs a multi-modal distribution for localization and uniquely associates each location with its most probable orientation. It avoids a dense search over all 3-DoF poses (localization+orientation) by discretizing the orientation sparsely and performing additional regression. This formulation is efficient for fine-grained pose estimation. It is also shown that the predicted probability can be used to filter out predictions that potentially have large localization or orientation errors. (iii) The proposed architecture exploits the strength of a translational equivariant feature encoder and contrastive learning. Its ground image encoder maintains the spatial scene layout information relative to camera's viewing direction in the ground image descriptor and the contrastive loss enforces aerial descriptors to encode global orientation information. These descriptors enable jointly localization and orientation estimation with negligible extra computational cost. Without re-training, the proposed model can infer the camera pose on images with different horizontal Field of Views (FoVs). In addition, it can utilize a coarse orientation prior, if available, to improve the localization without re-training.

4.2 Methodology

Given a ground-level color image G of size $H \times W \times 3$ and an aerial color image A of size $L \times L \times 3$ that covers the local surroundings of G, this chapter aims to estimate the 3 Degrees-of-Freedom (DoF) pose, $\hat{\mathbf{P}} \in \mathbb{R}^2 \times \$O(2)$, of the camera that took G. Specifically, $\hat{\mathbf{P}} = [\hat{x}, \hat{o}]$. $\hat{x} = (\hat{u}, \hat{v})$ denotes the image coordinates of the location of the camera of G in the aerial image A. $\hat{o} \in [0^\circ, 360^\circ)$ denotes the orientation of the camera in the 2D aerial image plane: 0° means heading North, i.e. the up direction in the aerial image, and the orientation angle increases in the clockwise direction. Similar to other cross-view camera pose estimation

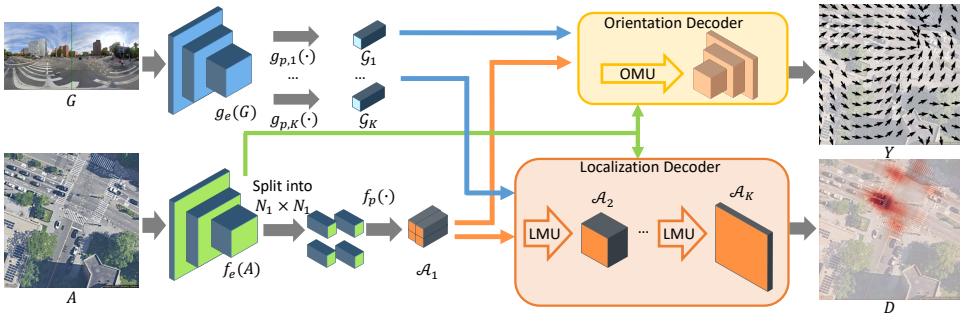


Figure 4.2: An overview of the proposed Convolutional Cross-View Pose Estimation method, CCVPE. The output localization distribution (in red) and orientation vector field (black arrows) are overlaid on top of the input aerial image for intuition.

methods [33, 130], it is assumed that the pitch and roll angle of the ground camera are small, which is often the case for a vehicle-mounted camera.

4.2.1 Methodological design considerations

Existing cross-view camera pose estimation methods [31, 127, 128] use a Siamese network with two image encoders without weight-sharing, fuse the encoder's descriptors at the bottleneck, and finally, have a decoder provide the output. The proposed model follows a similar approach with a few novel modifications.

1. Multi-modal prediction: Instead of treating localization as a uni-modal estimation problem [127, 128], this chapter proposes to predict location with a discrete probability distribution D over the pixels in the $L \times L$ aerial image A, and formulate the learning as multi-class classification. This way, the output can capture the potential multi-modal localization ambiguity, and assign high probability to multiple distinct aerial locations that match the observed ground image G. The probabilistic output could be provided to a downstream robot localization stack for fusion with other sensors, or the *Maximum A-Posteriori* (MAP) location can be taken as a single localization estimate. Furthermore, the probability provides a confidence estimate suitable to reject unreliable predictions, as the experiments will demonstrate.

2. Coarse-to-fine descriptor matching: To obtain a high-resolution localization distribution, this chapter proposes to *match* a single ground descriptor to local regions in the aerial feature map, e.g. using the cosine similarity. The concept of learning a shared feature space where descriptors from different views are compared is also encountered in cross-view image retrieval [105, 106, 118], but this chapter applies it for dense localization prediction. The proposed approach can therefore benefit from the contrastive learning loss to learn discriminative feature spaces for matching.

Furthermore, it is observed that the discriminative visual information that distinguishes one aerial region from another depends on the aerial resolution and scale. This chapter therefore proposes to apply this descriptor matching approach in a coarse-to-fine manner, starting at the low-resolution bottleneck, doubling the feature map resolution each time until the full target resolution is reached. At each subsequent level, the proposed approach

will match the ground and aerial information and use the resulting matching score to guide the upsampling of the aerial feature to a higher spatial resolution. Experiments will show that this improves localization accuracy.

3. Joint location and orientation matching: Location and orientation should be considered jointly. Estimating one, while ignoring the other could lead to sub-optimal estimation since the observed layout of the scene in the ground image G only relates to the BEV layout when both location and orientation of the ground camera are correct. Meanwhile, exploring a prior in one, e.g. orientation, should also benefit the estimation of the other, e.g. localization. This leads us to two considerations:

First, the image descriptors should *not* be invariant to different orientations. Instead, the proposed method constructs ground descriptors where the elements correspond to information for specific viewing directions *relative* to the camera's unknown orientation, and aerial descriptors where dimensions capture information in specific *global* viewing directions (see Figure 4.1 left). An aerial descriptor should only match the ground descriptor if the locations are similar, and if the viewing directions are aligned. By constructing ground descriptors that are *equivariant* with the camera's viewing direction (i.e. the horizontal image direction), the correct global orientation of the ground camera can be found by reordering its descriptor's feature dimensions ('rolling' the descriptors, see Figure 4.1 middle) to match the local aerial descriptor.

Second, in addition to the Localization Decoder, the proposed model adds an Orientation Decoder that predicts orientation as a function of the predicted location, i.e. it predicts a 2D vector field Y over the aerial view that maps each aerial location to the ground camera's most probable orientation if it would be located there. For instance, if the ground image shows the camera oriented towards a crossing, the localization uncertainty in the aerial view could be spread across the streets approaching the crossing, and each location would suggest a different global orientation (see Figure 4.1 right). Uncertainty in the localization output thus also captures uncertainty over the global orientation.

4. Generalize to different horizontal FoVs: This chapter aims for a model that can be used to match panoramic ground images, as well as images with a limited horizontal FoV without re-training, and can be trained with images of different FoVs for data augmentation. Therefore, other than constructing descriptors with a fixed length, the proposed ground descriptors have a flexible length that depends on the horizontal FoV of the ground image G.

4.2.2 Architecture overview

The design considerations from Section 4.2.1 motivate the proposed Convolutional Cross-View Pose Estimation (CCVPE) architecture, shown in Figure 4.2. One branch of the network, $g(\cdot)$, encodes the ground image G, and another branch, $f(\cdot)$, encodes the aerial image A. The descriptors from both encoders are matched in two specialized decoder branches: the Localization Decoder predicts the 2D spatial distribution D, the Orientation Decoder outputs the dense orientation vector field Y.

To match descriptors in a coarse-to-fine manner at K levels, the ground image G will be encoded into K ground descriptors $\mathcal{G}_k, k \in \{1, \cdots, K\}$, each of a different length C_k^G and capturing the relevant information to distinguish poses at that level's spatial resolution. Similarly, K aerial descriptor maps $\mathcal{A}_k$ are constructed to represent the relevant matching information of each local aerial region at level k. Each aerial descriptor $\mathcal{A}_k^{i,j}$ at spatial

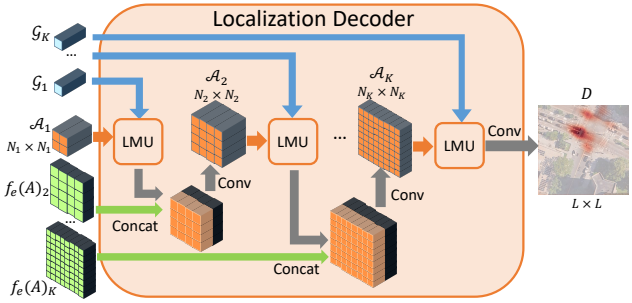


Figure 4.3: The proposed Localization Decoder.

4

location (i, j) in the descriptor map $\mathcal{A}_k$ has a length of C_k^A, which represents all 360° viewing directions at that local region. When the ground descriptor $\mathcal{G}_k$ is encoded from a 360° panoramic ground image, it similarly has $C_k^G = C_k^A$. If the ground image instead has a limited horizontal FoV, then $C_k^G < C_k^A$ and its descriptors will later be matched to only C_k^G of the C_k^A aerial descriptor dimensions. The spatial resolution of the aerial descriptor map at level k is $N_k \times N_k = 2N_{k-1} \times 2N_{k-1}$, where $N_1 \times N_1$ is the lowest resolution at the bottleneck, and $N_K \times N_K = L/2 \times L/2$ is the last matching level K before the final output. The first aerial descriptor map, $\mathcal{A}_1$, is shared between the Localization Decoder and the Orientation Decoder.

In the Localization Decoder, see Figure 4.3, the ground and aerial descriptors are compared at multiple resolution levels in a coarse-to-fine manner with the proposed novel Localization Matching Upsampling (LMU) module. In the Orientation Decoder, a similar Orientation Matching Upsampling (OMU) module is employed, though only once after the bottleneck (experiments will demonstrate this decoder does not benefit from coarse-to-fine matching). Similar to UNet [88] and other models for dense prediction tasks [151–153], this chapter furthermore adds skip connections from the aerial encoder to the two decoders between feature maps of same spatial resolution.

In the following, details on the proposed descriptor construction, descriptor matching modules, localization and orientation decoders, and used loss functions are provided.

4.2.3 Descriptors construction

Both ground and aerial encoders $g(\cdot)$ and $f(\cdot)$ first apply their own feature extractor, $g_e(\cdot)$ and $f_e(\cdot)$, respectively. For the ground branch, the proposed model then uses K ground feature projectors $g_{p,k}(\cdot), k \in \{1, \cdots, K\}$ to extract from the encoder's feature map the descriptors for the different coarse-to-fine levels, i.e. $\mathcal{G}_k = g_{p,k}(g_e(G))$. For the aerial branch, the proposed model splits the aerial feature volume $f_e(A)$ into $N_1 \times N_1$ sub-volumes and uses a shared aerial feature projector $f_p(\cdot)$ on each sub-volume $f_e(A)^{i,j}$ to generate the $N_1 \times N_1$ aerial descriptor map at level 1, $\mathcal{A}_1^{i,j} = f_p(f_e(A)^{i,j})$. The aerial descriptor maps $\mathcal{A}_k$ for $k > 1$ will be constructed within the Localization Decoder.

This chapter will assume that ground images follow a cylindrical projection, namely that each column of pixels in the image represents the same number of degrees in the horizontal FoV. While cylindrical projections are commonly used for panoramic images,

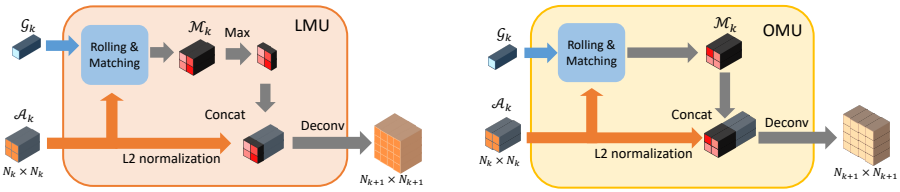


Figure 4.4: The proposed Localization Matching Upsampling (LMU) and Orientation Matching Upsampling (OMU) modules. Both generate an aerial feature map at a higher resolution than its input by matching the input aerial features to the ground features.

regular images with a limited horizontal FoV typically do not use this projection. This chapter still models all ground images as such, since it is found that this approximation still works well in practice. The proposed model uses a translational equivariant ground encoder $g(\cdot)$, therefore the length $C_k^G = C_k^A \times \frac{F}{360}$ of the ground descriptors $\mathcal{G}_k$ reflects the F degrees horizontal FoV of the ground image G.

Feature extractors: A regular convolutional network backbone is used as the ground and aerial feature extractors $g_e(\cdot)$ and $f_e(\cdot)$ on the input $H \times W \times 3$ ground image G and input $L \times L \times 3$ aerial image A, without sharing weights between these branches. This chapter denotes the shape of the encoded ground feature maps as $H' \times W' \times C'$, and of the aerial feature maps as $L' \times L' \times C'$.

Ground feature projector: A projector $g_{p,k}(\cdot)$ produces a single ground descriptor of length C_k^G. To reduce the computational cost of matching at increasingly higher spatial resolutions, the length of the ground descriptor at the next level is half of that of the ground descriptor at the current level, i.e. $C_k^G = 2C_{k+1}^G$.

Each projector consists of a 1×1 convolution to reduce the C' feature channels of the extracted ground feature $g_e(G)$ to $C'_k < C'$. To summarize the information along the vertical (height) direction in the scene while keeping it equivariant with the horizontal direction (relative viewing direction), the proposed model applies a fully-connected operation along the columns and squeezes the column dimension from H' to 1, resulting a $1 \times W' \times C'_k$ feature map. Finally, the ground descriptor $\mathcal{G}_k$ is created by reshaping this feature map into a 1D vector of length $C_k^G = W'C'_k$. These ground descriptors $\mathcal{G}_k$ are explicitly orientation-aware, as every block of C'_k elements captures the semantic content in a specific horizontal viewing direction relative to the camera's orientation.

Aerial feature projector: The proposed model creates spatial granularity for localization by splitting the $L' \times L' \times C'$ aerial feature volume into $N_1 \times N_1$ feature sub-volumes. Then, a shared fully connected layer is used as the aerial feature projector $f_p(\cdot)$ to map each of the sub-volumes into an aerial descriptor $\mathcal{A}_1^{i,j}$. The orientation awareness of the proposed aerial descriptors is encouraged by the proposed loss function, which will align the aerial descriptors with the orientation-aware ground descriptors, see details in Section 4.2.6.

4.2.4 Descriptor matching modules

To jointly consider location and orientation, the proposed model matches ground descriptors at different locations in the aerial image and considers R different global orientations. In particular, the matching is done inside the proposed descriptor matching modules, the

Localization Matching Upsampling (LMU) module, and the Orientation Matching Upsampling (OMU) module. As seen in Figure 4.4, both modules rely on a Rolling & Matching strategy to compute descriptor matching scores.

Rolling & Matching: Both LMU and OMU use the ground descriptor $\mathcal{G}_k$ to 'match' the aerial descriptors $\mathcal{A}_k$ at each candidate location (i, j) with a defined global orientation $\frac{r}{R}360^\circ, r \in \{1, \cdots, R\}$, and output a feature volume with the higher spatial resolution of the next level $k+1$. To create R global orientations, $[0^\circ, \frac{1}{R}360^\circ, \cdots, \frac{R-1}{R}360^\circ]$, the proposed model 'rolls' the orientation-aware aerial descriptor $\mathcal{A}_k^{i,j}$ at each candidate location (i, j) R times. Specifically, each 'rolling' is achieved by shifting all elements in $\mathcal{A}_k^{i,j}$ by a step length of $\frac{C_k^A}{R}$ to the front, and moving the C_k' front-most elements to the back. Note that R is selected such that the rolling step length $\frac{C_k^A}{R}$ is a multiple of C_k'. The resulting aerial descriptors $\mathcal{A}_k^{i,j,r}$ each represents 'what the ground descriptor at level k should contain' at a particular location and global orientation combination $(i, j, \frac{r}{R}360^\circ)$.

Matching each aerial descriptor $\mathcal{A}_k^{i,j,r}$ to the ground descriptor $\mathcal{G}_k$ is then done by the cosine similarity. Whereas each $\mathcal{A}_k^{i,j,r}$ captures the environment's appearance in all global directions with a C_k^A-dimensional vector, the C_k^G-dimensional ground descriptor $\mathcal{G}_k$ may represent images with a limited horizontal FoV, i.e. $C_k^G < C_k^A$. Therefore, the proposed model crops the middle C_k^G elements from $\mathcal{A}_k^{i,j,r}$, denoted as $\overline{\mathcal{A}}_k^{i,j,r}$, to match the same-sized descriptors of the same FoV using,

$$\mathcal{M}_k^{i,j,r} = sim(\mathcal{G}_k, \mathcal{A}_k^{i,j,r}) = \frac{\mathcal{G}_k \cdot \overline{\mathcal{A}}_k^{i,j,r}}{\|\mathcal{G}_k\|_2 \times \|\overline{\mathcal{A}}_k^{i,j,r}\|_2}. \tag{4.1}$$

The Rolling & Matching can be seen as convolving a kernel $\mathcal{G}_k$ over $\mathcal{A}_k^{i,j}$ with a stride of $\frac{C_k^A}{R}$, circular padding, and extra normalization. The resulting $N_k \times N_k \times R$ matching score volume $\mathcal{M}_k$ expresses how similar the ground descriptor is to the aerial descriptor at each candidate location and orientation.

LMU: The LMU summarizes the localization cues from $\mathcal{M}_k$ in an invariant manner to the different global orientations. The proposed model therefore takes for each location the maximum matching score over the R orientations. These $N_k \times N_k \times 1$ max scores are concatenated to the L2-normalized $N_k \times N_k$ aerial descriptors to guide the upsampling of the aerial feature through a deconvolution. It will be shown in the ablation study that the L2-normalization before feature concatenation is crucial for good pose estimation performance.

Notably, a prior in the ground camera's orientation is often available for vehicle localization, e.g. indicated by the driving direction. Incorporating such an orientation prior is straightforward in the LMU by removing the non-corresponding orientation channels in the matching score volume $\mathcal{M}_k$. This does not require any retraining.

OMU: Instead of extracting features that are orientation-invariant, OMU explicitly maintains the orientation information in $\mathcal{M}_k$. It has a similar design as LMU other than that the $\mathcal{M}_k$ is directly concatenated to the L2-normalized aerial descriptors. Thus the deconvolution layer can make use of information on how the ground descriptor $\mathcal{G}_k$ matches aerial descriptors $\mathcal{A}_k$ at all R orientations. The proposed method applies OMU only to

matching level 1 in the Orientation Decoder (other settings are also explored, but no clear benefit was observed).

4.2.5 Decoders

The proposed model has two separate decoders for localization and orientation estimation.

Localization Decoder: The proposed Localization Decoder contains K LMU modules to gradually increase the spatial resolution of ground and aerial descriptor matching and finally generates a discrete distribution D over the pixels of $L \times L$ aerial image A for localization, see Figure 4.3.

At each level k, the output feature from LMU is concatenated with the skip-connected aerial feature $f_e(A)_{k+1}$ of the same spatial resolution from the aerial feature extractor $f_e(\cdot)$ to access the scene layout information. Then, 2D convolution is applied to generate the aerial descriptors $\mathcal{A}_{k+1}$ for level $k+1$. After the LMU at level K, the output feature volume would have a spatial resolution $L \times L$, where $L = 2N_K$. Next, 2D convolution with a softmax activation is applied to convert the feature volume into a $L \times L \times 1$ discrete distribution D, in which the values denote how probable the ground camera is located at each pixel location (i, j). The *Maximum A-Posteriori* (MAP) pixel location $(\hat{i}, \hat{j})$ in D is taken as the final localization estimation, and the image coordinate of its center is the final prediction, $(\hat{u}, \hat{v}) = (u_{\hat{i}}, v_{\hat{j}})$.

Orientation Decoder: The proposed Orientation Decoder up-samples the coarse orientation information into a dense orientation vector field. It contains an OMU module at the beginning to match the ground descriptor $\mathcal{G}_1$ and aerial descriptors $\mathcal{A}_1^{i,j}$ at level 1 and upsamples the resulting matching score volume together with the L2-normalized aerial descriptors to spatial resolution $N_2 \times N_2$. The remainder of the Orientation Decoder uses a series of deconvolutions and convolutions to further upsample the feature volume to the target resolution $L \times L$. Similar to the proposed Localization Decoder, there is a skip connection that passes aerial feature $f_e(A)_k$ from the aerial encoder to the Orientation Decoder. The final output of the Orientation Decoder is an $L \times L \times 2$ vector field Y denoting the predicted orientation at each pixel location (i, j) in the aerial image A. The feature channel is L2-normalized and the first and second channels are used to represent the cosine and sine of the predicted orientation angle. The final orientation prediction $\hat{o}$ is selected in Y at the predicted pixel location $(\hat{i}, \hat{j})$, i.e. $\hat{o} = Y^{(\hat{i},\hat{j})}$.

4.2.6 Loss functions

The proposed loss $\mathcal{L}$ consists of three parts: a contrastive learning loss $\mathcal{L}_{\mathcal{M}}$, a classification loss $\mathcal{L}_D$ for localization, and a regression loss $\mathcal{L}_Y$ for orientation estimation. The ground truth location is represented by a discrete distribution D_{gt} of size $L \times L$. In practice, a 2D Gaussian distribution is placed at the ground truth pixel coordinates (i_{gt}, j_{gt}) to form a smooth ground truth distribution D_{gt}.

The contrastive learning loss: $\mathcal{L}_{\mathcal{M}}$ is an average over contrastive learning losses $\mathcal{L}_{\mathcal{M}k}$, at K levels, i.e. $\mathcal{L}_{\mathcal{M}} = \frac{1}{K}\sum_{k=1}^{k=K} \mathcal{L}_{\mathcal{M}k}$. At each level k, $\mathcal{L}_{\mathcal{M}k}$ is applied on the matching score volume $\mathcal{M}_k$ to encourage the aerial descriptors for locations and orientations close to the ground truth poses to match the ground descriptor $\mathcal{G}_k$. Since the proposed ground descriptors are orientation equivariant, training with $\mathcal{L}_{\mathcal{M}k}$ enforces the aerial descriptors at the correct locations to be orientation equivariant as well.

The location and orientation space of $\mathcal{M}_k$ is discretized into $N_k \times N_k \times R$, and the ground truth location and orientation would never exactly be centered at a grid point. This chapter therefore expresses the closeness of indices (i, j, r) at level k to the true pose with weights $w_k^{i,j,r} = w_k^{i,j} \times w_k^r$. To obtain spatial weights $w_k^{i,j}$, the spatial dimensions of D_{gt} is reduced from $L \times L$ to $N_k \times N_k$ by max pooling. To compute weights w_k^r over the R orientation channels, the orientation indices r_1 and r_2 of the discrete angles closest to the true orientation are found, $360^\circ \times (r_1/R) < o_{gt} < 360^\circ \times (r_2/R)$. The model only assigns non-zero weight to r_1 and r_2, where their weight is inversely proportional to their relative angular distance to o_{gt}, and $w_k^{r_1} + w_k^{r_2} = 1$. Finally, $\mathcal{L}_{\mathcal{M}k}$ is defined as a weighted sum $\mathcal{L}_{\mathcal{M}k} = \sum_{i,j,r} w_k^{i,j,r} \mathcal{L}'_{\mathcal{M}k}(i, j, r)$ of infoNCE losses [154] on the cosine similarity of Equation (4.1),

$$\mathcal{L}'_{\mathcal{M}k}(i, j, r) = -\log \frac{\exp(sim(\mathcal{G}_k, \mathcal{A}_k^{i,j,r})/\tau)}{\sum_{i',j',r'} \exp(sim(\mathcal{G}_k, \mathcal{A}_k^{i',j',r'})/\tau)}. \tag{4.2}$$

The localization loss: This chapter formulates the localization problem as a multi-class classification. In the main setting, the localization loss $\mathcal{L}_D$ is a cross-entropy loss,

$$\mathcal{L}_D = -\sum_{i=1}^{L} \sum_{j=1}^{L} D_{gt}^{i,j} \log D^{i,j}, \tag{4.3}$$

where (i, j) are pixel coordinates.

As an alternative, it is also considered to train the localization distribution D by minimizing the transported mass from D to D_{gt} based on Optimal Transport theory [155]. For this, a Wasserstein distance-based loss [156] is used as the $\mathcal{L}_D$. Unlike cross-entropy, Wasserstein distance considers the distance between the mass in the source and target distributions. To compute the Wasserstein distance loss $\mathcal{L}_D$ between D and D_{gt} efficiently, D_{gt} is defined as a one-hot distribution. This loss then becomes,

$$\mathcal{L}_D = -\sum_{i=1}^{L} \sum_{j=1}^{L} d(i, j) \cdot D^{i,j}. \tag{4.4}$$

In Equation (4.4), $d(i, j)$ denotes the L2-distance in pixels between the pixel location (i, j) in D to the ground truth pixel location (i_{gt}, j_{gt}), i.e. $d(i, j) = \sqrt{(i - i_{gt})^2 + (j - j_{gt})^2}$. The cross-entropy loss and Wasserstein distance-based loss are compared in the ablation study, but the cross-entropy loss will be used as $\mathcal{L}_D$ in the main experiments.

The orientation loss: Instead of treating the orientation prediction as a discrete classification problem, which would result in a large number of classes for joint localization and orientation supervision, this chapter formulates this problem as regression. Since this chapter uses a Gaussian smoothed ground truth D_{gt}, the contributions from smoothed ground truth locations are summed and the proposed orientation loss $\mathcal{L}_Y$ is defined as,

$$\mathcal{L}_Y = \sum_{i=1}^{L} \sum_{j=1}^{L} D_{gt}^{i,j} \left(\left(\cos\left(o_{gt}\right) - Y_1^{i,j} \right)^2 + \left(\sin\left(o_{gt}\right) - Y_2^{i,j} \right)^2 \right). \tag{4.5}$$

In Equation (4.5), o_{gt} is the ground truth orientation, Y is the $L \times L \times 2$ predicted orientation vector field. Y_1 and Y_2 denote the first and second channel of Y. Multiplying with D_{gt} removes the contribution to the orientation loss $\mathcal{L}_Y$ at wrong locations.

The total loss: $\mathcal{L}$ is a weighted combination of the localization loss $\mathcal{L}_D$, orientation loss $\mathcal{L}_Y$, and contrastive learning loss $\mathcal{L}_M$:

$$\mathcal{L} = \mathcal{L}_D + \alpha \mathcal{L}_Y + \beta \mathcal{L}_M, \tag{4.6}$$

where the α and β are hyperparameters that weigh the importance of $\mathcal{L}_Y$ and $\mathcal{L}_M$ during training.

4.3 Experiments

This section will first introduce the three used datasets, followed by the evaluation metrics. After this, the implementation details are provided. Then, the proposed CCVPE method is compared to previous state-of-the-art baselines w.r.t. generalization to new measurements within the same areas and across different areas. Next, this section studies how the proposed method works with an orientation prior and on images with different horizontal FoVs. Then, the proposed method is used to estimate the pose of the ego-vehicle along test traversals using sequences of ground images. Finally, an extensive ablation study and an analysis of runtime are provided.

4.3.1 Datasets

This section tests the generalization of all models to new measurements within the same areas and across different areas on the VIGOR [127] and KITTI [20] datasets. On the Oxford RobotCar dataset [1, 145], the proposed method is used to estimate the ego-vehicle pose frame-by-frame using the ground image sequence collected in test traversals. The original KITTI and Oxford RobotCar datasets do not contain any aerial images, therefore this chapter makes use of the collected aerial images from [110] for KITTI, and [30, 31] for Oxford RobotCar.

The VIGOR dataset [127] contains ground-level panoramic images and aerial patches collected in four cities in the US. The aerial patches are distributed regularly as a grid, providing seamless coverage of the 4 target cities. Each aerial patch covers a ~70 m × 70 m ground region. The orientation of the panorama and aerial patch is aligned such that the center vertical line in the panorama corresponds to the up direction (North) in the aerial patch. In our experiments, changing the orientation of the ground panorama is achieved by shifting the image along the horizontal axis. Reducing the horizontal FoV is achieved by dropping the image columns at the left and right borders. Since the ground truth labels are improved by [33], this dissertation uses those more accurate labels. The VIGOR dataset defines the aerial patches as either positive or semi-positive for each ground image. An aerial patch is positive if its center 1/4 region contains the ground camera's location, otherwise, it is semi-positive. In the following experiments, positive aerial images are used for training and testing all models. The experiments adopt the Same-Area and Cross-Area split from [127]. On the Same-Area split, models are trained on images from all four cities and tested on images from the same cities. Training and test sets do not

share any ground images but may share aerial patches. On the Cross-Area split, models are trained on image pairs from New York and Seattle and tested on pairs from Chicago and San Francisco. For validation and hyperparameter tuning, this chapter randomly selects 20% of the data from the training set.

The KITTI dataset [20] is collected by a vehicle platform in Karlsruhe, Germany, covering city, rural area, and highway scenarios. The stereo camera faces the driving direction and has a horizontal FoV of 90°. In [130], the authors make use of the images from the left camera of the stereo camera and collected aerial images with ground resolution ~0.20 m/pixel to enable cross-view pose estimation. Each aerial patch covers a ~100 m × 100 m ground area. The data is split into Training, Test 1, and Test 2 sets. Images in Training and Test 1 sets are from the same regions. Images in Test 2 set are from different areas than those in the Training set. In the experiments, Test 1 and Test 2 sets are named as Same-Area and Cross-Area. As assumed in [130], ground images are located within a 40 m × 40 m area in the corresponding aerial patches' center, and there is an orientation prior with noise between −10° and 10°. In this case, a random rotation between −10° and 10° is applied on each aerial image whose 'East' orientation was aligned with the ground image. In the experiments, this chapter adopts the same setting, and also provides extra results for unknown orientations where a random rotation from the 360° circular domain is applied on each orientation-aligned aerial image.

The Oxford RobotCar dataset [1, 145] contains videos with a limited horizontal FoV collected over multiple traversals at different times, seasons, and weather conditions, along the same route in Oxford, UK. In [29, 30], the authors collected aerial patches for retrieval, and later [31] stitched those aerial patches with their collected extra ones into a continuous aerial image that covers the Oxford area. This chapter follows the same setting as in [30, 31] that the training, validation, and test data are from different traversals to test our model's generalization to different dynamic objects, weather, and lighting conditions across time. Instead of directly using the sparse test images used in [30, 31], test ground images are sampled from the original Oxford RobotCar dataset [1, 145] at a higher frame rate, ~ 1.6 FPS, for our experiment of ego-vehicle following. In total, there are three test traversals, enabling testing in Summer and Winter. During training, aerial patches that cover ~74 m × 74 m ground area are randomly cropped from the continuous map around a location that is less than ~26 m away from the vehicle's location. For validation and testing, the same set of aerial patches as in [31] are used. In the experiment, the orientation of the ground camera is always assumed unknown, so this chapter simply uses the ground images and north-aligned aerial images as input pairs.

4.3.2 Baselines methods

The proposed CCVPE is compared to two types of baselines.

First, the state-of-the-art (SOTA) cross-view pose estimation baselines are included: the cross-view regression method (CVR) [127], iterative optimization method LM [130], and SliceMatch [33]. CVR [127] is originally designed for joint image retrieval and location regression. For a fair comparison, this chapter trains it for localization within a given aerial image (it is found that it achieves better localization error than training it for retrieval + localization). This chapter also trained a CVR model using the same EfficientNet-B0 [157] as its feature extractor, denoted as Eff-CVR. LM [130] uses an iterative method to estimate

the location and orientation of the ground camera on the aerial image. This dissertation uses the provided model by its author and the same setting on its prior [130] on KITTI dataset, namely, the ground images are located in a $40m \times 40m$ area in the input aerial image center, and a rough orientation prior with noise between -10° and 10° is available during training and test time. For completeness, the model trained and tested without any orientation prior is also included.

On the KITTI dataset, [110] evaluated several image retrieval or retrieval with orientation estimation baselines by limiting their searching area to a region of $40m \times 40m$. This chapter includes the same fine-grained cross-view image retrieval baselines from [110].

4.3.3 Evaluation metrics

The mean and median error over all test samples are used as the main evaluation metrics for both localization and orientation prediction. For localization, the error is the distance in meters between the predicted location and the ground truth location. For orientation, the error is the angular difference (°) between the predicted camera orientation at the predicted location and ground truth camera orientation. In addition, this chapter reports the percentage of test samples that have an error below certain thresholds, namely 1 m, 3 m, and 5 m for localization, and 1°, 3°, and 5° for orientation. For localization, longitudinal and lateral error w.r.t. the vehicle's driving direction are given separately [130]. For image retrieval methods, the localization error is calculated by measuring the distance between the center of the retrieved aerial image patch and the ground truth location.

To measure if the true location receives probability mass, and thus would not be discarded if used in a probabilistic temporal filter, the predicted probability at the ground truth pixel is also measured. For the baseline method that regresses [127] to a single location without uncertainty estimates, it is assumed that their prediction is the peak of an isotropic Gaussian distribution, and estimate the standard deviation of this Gaussian distribution on the validation set. SliceMatch [33] measures descriptors' similarity scores on their candidate poses. This chapter uses the scores at the candidate location that is closest to the ground truth location to derive the predicted probability at the ground truth pixel. Finally, we measure our model's runtime on a Tesla V100 GPU.

4.3.4 Implementation details

EfficientNet-B0 [157] is used as our ground and aerial feature extractors, $g_e(\cdot)$ and $f_e(\cdot)$. There is no weight-sharing between them. When the ground image G is panoramic, circular padding in the horizontal direction is used inside the ground encoder $g(\cdot)$, and zero padding otherwise. For other model components, and the vertical direction padding in $g(\cdot)$, zero padding is used. During training, the feature extractor is initialized from ImageNet [146] pre-trained weights, and other components are initialized randomly. The proposed model is trained using the Adam optimizer [147] with a learning rate of 0.0001. This chapter uses the default drop connect [158] rate, 0.2, from EfficientNet [157], and the default $\tau = 0.1$ in Eq. (4.2), from infoNCE loss [154]. Different weights $\alpha = 1 \times 10^{-2}, \cdots, 1 \times 10^{2}$ and $\beta = 10, \cdots, 1 \times 10^{5}$ were tested for weighing the orientation loss $\mathcal{L}_Y$ and contrastive learning loss $\mathcal{L}_M$, and $\alpha = 10$ and $\beta = 1 \times 10^{4}$ are selected since they provide best validation performance. The model bottleneck size $N_1 \times N_1$ is set to 8×8, and consequently there are $K = 6$ levels in our coarse-to-fine descriptor matching.

Note that, even though the proposed method assumes ground images follow cylindrical projection, for KITTI and Oxford RobotCar datasets this chapter directly inputs their perspective images. The experiment shows that both projections work equally well in practice.

4.3.5 Generalization to new measurements in same area

First, this chapter compares the proposed method, CCVPE, to baselines for generalizing to new measurements (i.e. ground images) in the same area. This corresponds to use cases that target operation in a predetermined area, such as driving in one city, so models can be trained on data from that specific area. For this task, this chapter reports the evaluation results on VIGOR Same-Area test set and KITTI Same-Area (Test 1) set.

Table 4.1: Evaluation on VIGOR Same-Area and Cross-Area test set. **Best in bold.** This chapter reports mean and median localization and orientation error, as well as the probability at the ground truth pixel in the aerial image, denoted as 'P@GT'. The left-most column indicates orientation uncertainty: '0°' means testing with known orientation, such that the center vertical line in the panorama corresponds to the North direction in the aerial image, and the known orientation is used to remove the non-corresponding orientation channels in the matching score volume; '360°' means the test orientation is unknown and then the panoramic ground image was horizontally shifted corresponding to a random angle in the 360° circular domain.

	VIGOR test	Same-Area					
		↓ Localization (m)		↓ Orientation (°)		↑ P@GT (1×10^{-3})	
		mean	median	mean	median	mean	median
0°	CVR [127]	8.82	7.68	-	-	0.02	0.02
0°	Eff-CVR	7.89	6.25	-	-	0.02	0.03
0°	SliceMatch [33]	5.18	2.58	-	-	0.06	0.05
0°	*CCVPE* (proposed)	**3.60**	**1.36**	-	-	**1.60**	**1.12**
360°	SliceMatch [33]	8.41	5.07	28.43	**5.15**	0.02	0.02
360°	*CCVPE* (proposed)	**3.74**	**1.42**	**12.83**	6.62	**1.47**	**1.00**
	VIGOR test	Cross-Area					
		↓ Localization (m)		↓ Orientation (°)		↑ P@GT (1×10^{-3})	
		mean	median	mean	median	mean	median
0°	CVR [127]	9.45	8.33	-	-	0.02	0.02
0°	Eff-CVR	8.27	6.60	-	-	0.02	0.03
0°	SliceMatch [33]	5.53	2.55	-	-	0.06	0.06
0°	*CCVPE* (proposed)	**4.97**	**1.68**	-	-	**1.08**	**0.71**
360°	SliceMatch [33]	8.48	5.64	**26.20**	**5.18**	0.02	0.02
360°	*CCVPE* (proposed)	**5.41**	**1.89**	27.78	13.58	**0.93**	**0.58**

Pose estimation on VIGOR Same-Area: As shown in Table 4.1 Same-Area, when testing with images with known orientation, our method surpasses all baselines, CVR [127], Eff-CVR, and SliceMatch [33], w.r.t. mean and median localization errors. Replacing the VGG [159] backbone with EfficientNet-B0 [157] for CVR improves localization performance, but Eff-CVR still has significantly higher localization errors than ours. When the orientation of test images is unknown, our method beats the previous SOTA SliceMatch by a large margin in localization, i.e. 56% in the mean error and 72% in the median error. Regarding orientation estimation, CVR could not infer the orientation of the ground camera, and

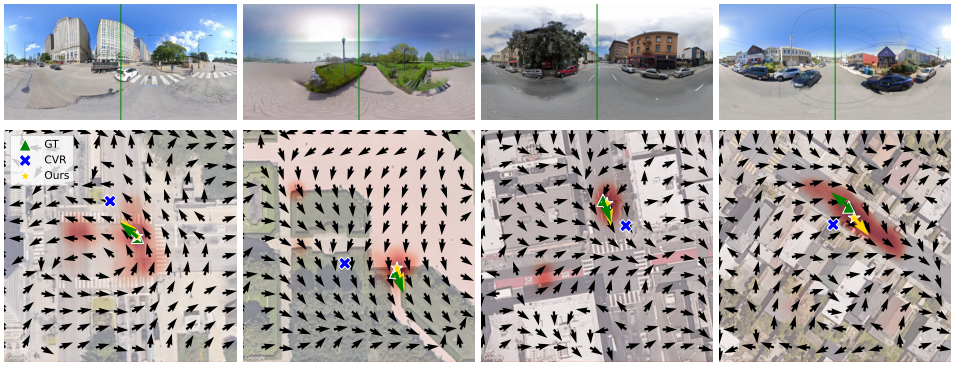


Figure 4.5: Qualitative results on VIGOR. First two samples are from the Same-Area test set, and the last two samples are from the Cross-Area test set. The first three samples are success cases, the fourth shows a failure case. CVR (blue cross) receives orientation-aligned ground and aerial images and does not estimate the orientation. CCVPE (ours) selects the orientation (yellow arrow) from the prediction location (yellow star) and dense orientation map Y (black arrows). The red color shows the localization probability distribution, and the darker the color the higher the probability. The center of the ground image is always the forward direction (green vertical line), which aligns with the true orientation in the aerial view (green arrow).

thus it is not included in the comparison. SliceMatch and our method address orientation prediction differently. SliceMatch divides the 360° orientation space into 64 bins and selects the most probable one based on descriptors matching, while our method creates $R = 20$ orientation scores and regress the true orientation after the grid-based matching. Quantitatively, our method has a lower mean orientation error but a slightly higher median orientation error than SliceMatch. Because of the regression formulation, the proposed method can smoothly track the change in the orientation of the ground camera. Grid-based solutions, such as SliceMatch, would need to densify their grid, resulting in more memory and computation needs.

Pose estimation on KITTI Same-Area: As shown in Table 4.2 Same-Area, camera pose estimation methods, LM [130], SliceMatch [33], and ours, outperform image retrieval-based methods in terms of percentages of test samples with lateral and longitudinal errors within the given thresholds. When a ±10° orientation prior is considered in both training and testing, as assumed in [130], our method has a lower mean/median error for both localization (1.22 m / 0.62 m) and orientation estimation (0.67°/0.54°) than LM and SliceMatch.

LM needs an orientation prior to guarantee there is an overlap in the scene between its projected aerial view and the ground view for iterative optimization. As a result, LM does not work when such an orientation prior is absent, see Table 4.2. Under this more challenging setting, the performance of both SliceMatch and our method degenerates. Our model still surpasses SliceMatch in localization performance but our model has higher errors in orientation estimation.

Qualitative results: Compared to single-mode estimators, e.g. regression-based CVR [127] and iterative optimization-based LM [130], our model shows its advantage especially when the scene contains a symmetric layout. As shown in the first two samples in Figure 4.5, when there are multiple visually similar locations, e.g. zebra crossings or

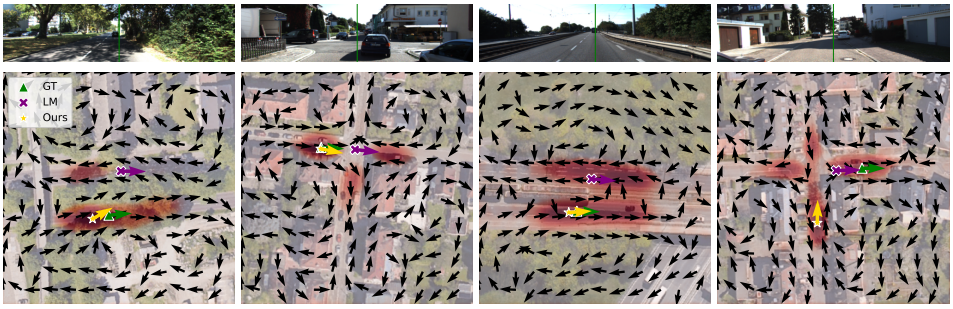


Figure 4.6: Qualitative results on KITTI. First two samples are from the Same-Area test set, and the last two samples are from the Cross-Area test set. The first three samples are success cases, the fourth one shows a failure case. An orientation prior with $\pm 10^{\circ}$ noise is provided to LM model, and the proposed model does not use an orientation prior.

junctions, CVR regresses to a location between them. The iterative method LM sometimes converges to a wrong local optimum, e.g. another road, see Figure 4.6. Our model expresses its uncertainty with its multi-modal distribution to capture all probable modes, usually identifying the correct location in all datasets.

Benefiting from our joint consideration of localization and orientation, when there are multiple probable locations in our prediction, our model predicts for each of these locations the most likely orientation. As shown in the first image pair in Figure 4.5, the orientation of the camera (the green line in the ground image) roughly points to the end of a zebra crossing. Our prediction suggests multiple locations, and each location predicts an orientation pointing to an end of a different zebra crossing.

At locations with a low localization probability, the ground-aerial descriptor matching has low similarity scores in all orientation channels. In this case, the orientation prediction is influenced less by the descriptor matching score but appears to follow a learned prior from the aerial view. Importantly, it will be shown in the ablation study that explicitly providing the descriptor matching scores is still key to good orientation prediction.

Probabilistic prediction: This section evaluates the probability estimation of the baselines and the proposed model on the VIGOR dataset. CVR [127] and Eff-CVR regress to a single location without any probability estimation. This dissertation fits a zero-mean Gaussian distribution on their predicted errors. The standard deviation of this Gauss is calculated based on their localization error on the validation set.

As shown in Table 4.1, our model has considerably higher mean and median probability at the ground truth location than CVR [127], Eff-CVR, and SliceMatch [33]. During training, SliceMatch is optimized for discriminative descriptors, while our model is directly optimized for high probability at the ground truth location by our cross-entropy loss. In general, our model is less likely to miss the ground truth location, which is an important aspect when the outputs are temporal filtered or fused with other sensor measurements.

Importantly, our probabilistic output can be used to identify predictions that potentially have large localization and orientation errors. Because the proposed method constructs orientation-aware descriptors, the better an aerial descriptor matches the ground descriptor, the more likely both the location and orientation of that aerial descriptor are correct. There-

Table 4.2: Evaluation on KITTI dataset. **Best in bold.** This chapter reports mean and median localization and orientation error, as well as the percentage of test samples that have lateral/longitudinal localization or orientation error less than a threshold. The evaluation of image retrieval methods (noted with 'retrieval') is collected from [110]. In the leftmost column, $\pm 10^\circ$ denotes training and testing with an orientation prior with noise in the range $[-10^\circ, 10^\circ]$, and 360° means no orientation prior by using noise from the 360° circular domain.

	Same-Area	↓ Loc. (m)		↑ Lateral (%)			↑ Longitudinal (%)			↓ Ori. (°)		↑ Ori. (%)		
		mean	med.	1m	3m	5m	1m	3m	5m	mean	med.	1°	3°	5°
retrieval	CVM-Net [105]	-	-	5.83	17.41	28.78	3.47	11.18	18.42	-	-	-	-	-
	CVFT [113]	-	-	7.71	22.37	36.28	3.82	11.48	18.63	-	-	-	-	-
	SAFA [106]	-	-	9.49	29.31	46.44	4.35	12.46	21.10	-	-	-	-	-
	Polar-SAFA [106]	-	-	9.57	30.08	45.83	4.56	13.01	21.12	-	-	-	-	-
	DSM [121]	-	-	10.12	30.67	48.24	4.08	12.01	20.14	-	-	3.58	13.81	24.44
$\pm 10^\circ$	LM [110]	12.08	11.42	35.54	70.77	80.36	5.22	15.88	26.13	3.72	2.83	19.64	51.76	71.72
	SliceMatch [33]	7.96	4.39	49.09	91.76	98.52	15.19	49.99	57.35	4.12	3.65	13.41	42.62	64.17
	CCVPE (proposed)	**1.22**	**0.62**	**97.35**	**98.65**	**99.71**	**77.13**	**96.08**	**97.16**	**0.67**	**0.54**	**77.39**	**99.47**	**99.95**
360°	LM [110]	15.51	15.97	5.17	15.13	25.44	4.66	15.00	25.39	89.91	90.75	0.61	1.88	2.89
	SliceMatch [33]	9.39	5.41	39.73	**80.56**	**87.92**	13.63	40.75	49.22	**8.71**	**4.42**	**11.35**	**36.23**	**55.82**
	CCVPE (proposed)	**6.88**	**3.47**	**53.30**	77.63	85.13	**25.84**	**55.05**	**68.49**	15.01	6.12	8.96	26.48	42.75

	Cross-Area	↓ Loc. (m)		↑ Lateral (%)			↑ Longitudinal (%)			↓ Ori. (°)		↑ Ori. (%)		
		mean	med.	1m	3m	5m	1m	3m	5m	mean	med.	1°	3°	5°
retrieval	CVM-Net [105]	-	-	6.96	21.55	35.24	3.58	10.45	17.53	-	-	-	-	-
	CVFT [113]	-	-	7.20	22.05	36.21	3.63	11.11	18.46	-	-	-	-	-
	SAFA [106]	-	-	9.15	27.83	44.27	4.22	11.93	19.65	-	-	-	-	-
	Polar-SAFA [106]	-	-	10.02	29.09	46.19	3.82	11.87	19.84	-	-	-	-	-
	DSM [121]	-	-	10.77	31.37	48.24	3.87	11.73	19.50	-	-	3.53	14.09	23.95
$\pm 10^\circ$	LM [110]	12.58	12.11	27.82	59.79	72.89	5.75	16.36	26.48	3.95	3.03	18.42	49.72	71.00
	SliceMatch [33]	13.50	9.77	32.43	78.98	86.44	8.30	24.48	35.57	4.20	6.61	46.82	46.82	46.82
	CCVPE (proposed)	**9.16**	**3.33**	**44.06**	**81.72**	**90.23**	**23.08**	**52.85**	**64.31**	**1.55**	**0.84**	**57.72**	**92.34**	**96.19**
360°	LM [110]	15.50	16.02	5.60	16.02	25.60	5.64	15.86	25.76	89.84	89.85	0.60	1.60	2.65
	SliceMatch [33]	14.85	11.85	**24.00**	**62.52**	**72.89**	7.17	26.11	33.12	**23.64**	**7.96**	**31.69**	**31.69**	**31.69**
	CCVPE (proposed)	**13.94**	**10.98**	23.42	49.15	60.46	**11.81**	**29.85**	**42.12**	77.84	63.84	3.14	9.24	14.56

fore, the localization probability can be used to filter the orientation error as well. As shown in Figure 4.7, when this chapter ranks the predictions based on their predicted probabilities, the more confident predictions have in general lower localization and orientation errors. This property is important in safety-crucial applications such as autonomous driving.

4.3.6 Generalization to new measurements across areas

Here, this section considers use cases that target operations in areas that were not covered specifically by the training data, such as driving in different cities or suburban areas.

Overall, it has been seen a similar trend in model comparison in the Cross-Area setting as in the Same-Area setting. On VIGOR Cross-Area test set, see Table 4.1, our model surpasses the previous SOTA SliceMatch [33] in localization by a large margin. When orientation is unknown, our median error is 66% lower than that of SliceMatch. However, our orientation error is higher than that of SliceMatch. On KITTI Cross-Area test set, see Table 4.2, when an orientation prior with $\pm 10^\circ$ noise presents during training and testing, our model surpasses both LM [130] and SliceMatch [33] in both localization and orientation. Without this prior, our model has lower mean and median localization error than SliceMatch, but our orientation error is higher.

Unsurprisingly, compared to the performance on the Same-Area test set, there is a performance degradation for all models. Our model could learn priors from the scene layout in the aerial image to guide its predictions. It becomes more challenging when the test aerial images are unseen. Since our predicted orientation is selected at the predicted

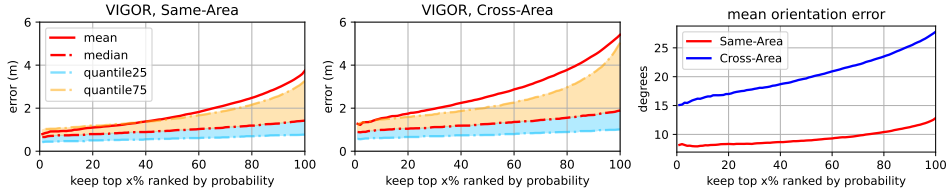


Figure 4.7: Ranking CCVPE's predictions based on their estimated probabilities (tested with unknown orientation). The more confident the prediction, the lower the localization and orientation error.

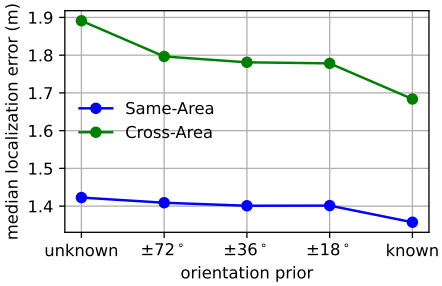


Figure 4.8: Localization with different orientation priors on VIGOR Same-Area and Cross-Area test set.

location, the orientation error is also likely to be large when localization is wrong, see Figure 4.6 last sample. It is also observed that there are more samples that have predicted orientation in the opposite direction on the Cross-Area test set than the Same-Area test set on both VIGOR and KITTI datasets. See the last sample of Figure 4.5 for an example. In practice, when there is a prior in orientation, e.g. identified by the driving direction, our model can make use of the prior to improve its prediction without retraining. This will be demonstrated in the next sub-section.

4.3.7 Effects on orientation prior and image's FoV

Next, this section studies the proposed model's behavior on both VIGOR Same-Area and Cross-Area test sets for inference with an orientation prior and ground images with different horizontal FoV.

Inference with an orientation prior: As described in Section 4.2.4, our model can make use of an orientation prior without retraining. Figure 4.8 shows that when a more accurate orientation prior is present, the localization performance increases accordingly. When there are multiple locations in the aerial image that match the ground image with different orientations, for example, at a crossroad, providing such an orientation prior effectively reduces the wrong matchings in our LMU and OMU modules, see examples in Figure 4.9.

Inference on images with different FoVs: Moreover, the proposed CCVPE model can infer on test ground images with other horizontal FoVs than in the training data. Figure 4.10 shows the results of our models trained with ground images with various horizontal FoVs being tested on ground images with different horizontal FoVs. In general, when the FoV of the ground image increases, the information contained in the image also increases. As a

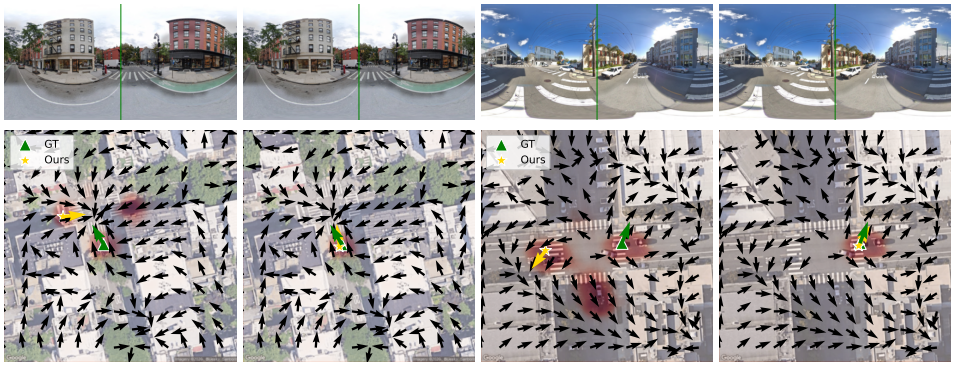


Figure 4.9: An orientation prior improves our localization performance on VIGOR, Same-Area (first two image pairs) and Cross-Area (last two image pairs). The first and third image: inference without an orientation prior. The second and fourth image: inference with an orientation prior containing noise between -36° and 36°. With the prior, locations that expect a different orientation become improbable.

result, it is seen a monotonic decrease in localization error when the test FoV increases for all models in Figure 4.10.

An example of the predictions from the proposed model trained with panoramic images is shown in Figure 4.11. When the FoV of the test ground image is 108° or 180°, the proposed model cannot distinguish different roads based on the limited content captured by the ground image, and thus predicts a multi-modal distribution to capture the probable locations. However, the peak of the distribution is in the wrong mode and consequently, the selected orientation is also wrong. When the test FoV increases to 252°, the peak of the output distribution is close to the ground truth location. Further increasing the FoV reduces the localization uncertainty and improves the localization. Notably, the proposed model can always access the full scene layout information from the aerial view no matter what the FoV of the ground view is. This example shows that the learned prior from the BEV layout solely is not enough for pose estimation, and our ground-aerial descriptor matching is crucial.

Because of the domain shift, the model trained with panoramas performs worse on images with small FoVs, compared to the model trained with images with a small FoV, see Figure 4.10. Besides, it is also observed that a steeper decrease in localization performance when the test FoV reduces for the model trained with panoramas than the model trained with images of a horizontal FoV of 108°. Training with images with a large FoV allows the model to use features that span widely in the ground image. When those features are absent, e.g. testing with a small FoV, the performance degenerates. On the other hand, if the training images only have a small FoV, the model would not learn to use features that span wider than the FoV of the training images. Consequently, increasing the test FoV brings less benefit. To tackle this trade-off, one can train the model with images whose FoVs are randomly sampled by cropping the panorama, e.g. sampled from $108^\circ, \cdots, 360^\circ$. Consequently, the resulting model performs well for all tested FoVs. Interestingly, this model also has slightly better localization performance than the model trained with images with FoV of 108° when inference on images with FoV of 108°. Note that this model is not

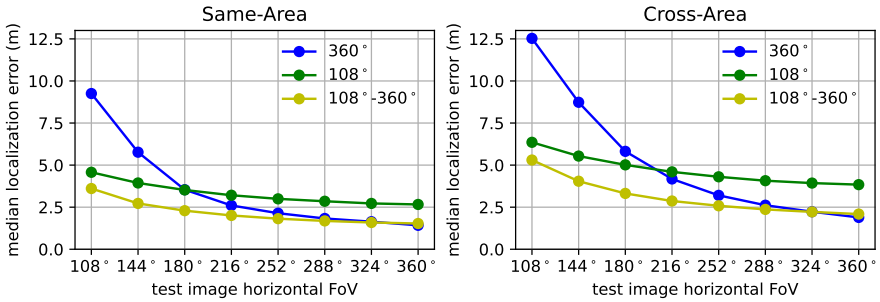


Figure 4.10: Localization on images with varying horizontal FoVs on VIGOR same-area and cross-area test sets. Green/blue/yellow curves represent the proposed model trained with horizontal FoV of 108°/360°/between 108° and 360°.

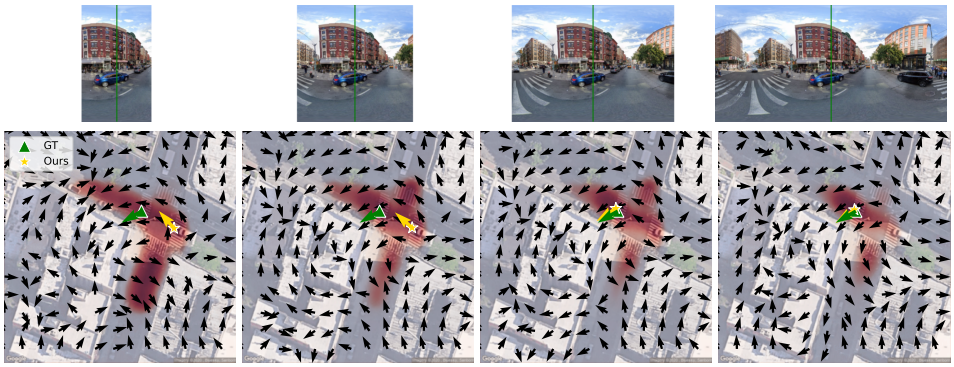


Figure 4.11: CCVPE model (ours) trained with panoramic image and inference on ground images with different FoVs (from left to right: 108°, 180°, 252°, 360°) from VIGOR samearea test set. Its dense orientation prediction is shown by black arrows. The final predicted orientation of the ground image (green vertical line) is shown by the yellow arrow.

used in our earlier comparison to other baselines for fairness since the baselines cannot include a similar data augmentation.

4.3.8 Ego-vehicle pose estimation across time

On the Oxford RobotCar dataset, the proposed model is deployed to follow the ego-vehicle over a sequence of ground-level images taken by the vehicle-mounted camera. To process a pair of input ground and aerial images on Oxford RobotCar, CCVPE takes 0.07 seconds, i.e. 14 FPS. It is assumed that there is a rough GNSS prior that identifies which aerial patch contains the location of the ground-level image. As shown in Figure 4.12, on all three test traversals, our model achieves median lateral and longitudinal localization error below 1 meter and median orientation error around 1°. Notably, even though the ground images in the Oxford RobotCar dataset have a small horizontal FoV compared to the panoramas in the VIGOR dataset, our model generalizes better to new ground images along the same route across time on the former dataset than to new panoramas on the latter dataset. On the Oxford RobotCar dataset, the aerial view can provide a strong prior as the vehicle

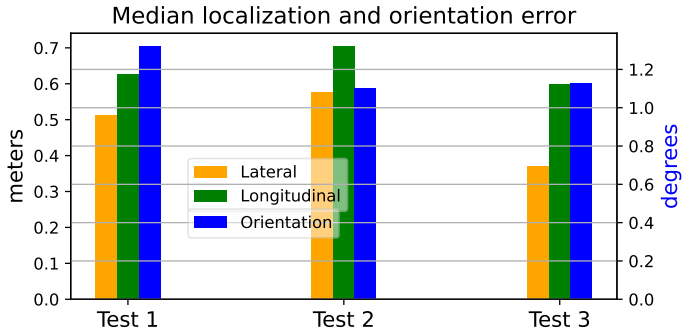


Figure 4.12: Median lateral and longitudinal localization error and median orientation error on Oxford RobotCar Test 1, 2, and 3 traversals.

always drives along the same route and the test area is seen during training. In contrast, the panoramas in the VIGOR dataset are not always captured on the road, plus its scenes are more diverse because of its broad coverage.

On the Oxford RobotCar sequences, it can be observed that how the predicted orientation map adapts to ground images at different nearby locations within the same aerial view, as seen in Figure 4.13. While the model learned a prior from the aerial view on the driving direction of the roads, the orientation predictions do respond to the ground image content at the high-probability locations. E.g. when the vehicle is in area A, the local orientation field points towards the junction seen in the ground view. The orientations in area A reflect a different orientation (a prior) once the vehicle moved on to area B.

4.3.9 Ablation study

Next, this section presents an ablation study on the VIGOR Same-Area validation set.

Number of LMU modules: As mentioned in Section 4.3.4, our model has $K = 6$ LMU modules for coarse-to-fine descriptor matching for localization. Here, this section studies the effect of LMU modules on localization performance by removing them at low or high levels. When removing an LMU module, the corresponding convolutional layer in the Localization Decoder is modified such that it directly processes the aerial feature without any matching scores.

Table 4.3: Effect of LMU modules on mean localization error on VIGOR same-area validation set. **Best in bold.**

VIGOR Same-Area, validation set											
K =	1	1,2	1,2,3	1,2,3,4	1,2,3,4,5	6	5,6	4,5,6	3,4,5,6	2,3,4,5,6	1,2,3,4,5,6
median error (m)	2.58	1.89	1.58	1.45	1.44	2.68	1.99	1.58	1.47	**1.42**	**1.42**

As shown in Table 4.3, the model with LMU modules at all 6 levels outperforms other variants in terms of the median localization error. When excluding the LMU modules at low or high levels, it is seen a consistent decrease in localization performance. Importantly, using LMU modules only at high levels, e.g. $K = 6$, does not provide equally good localization performance as the models that also have LMU modules at lower levels. Directly contrasting aerial descriptors at a fine resolution is a difficult learning task. Using LMU modules at

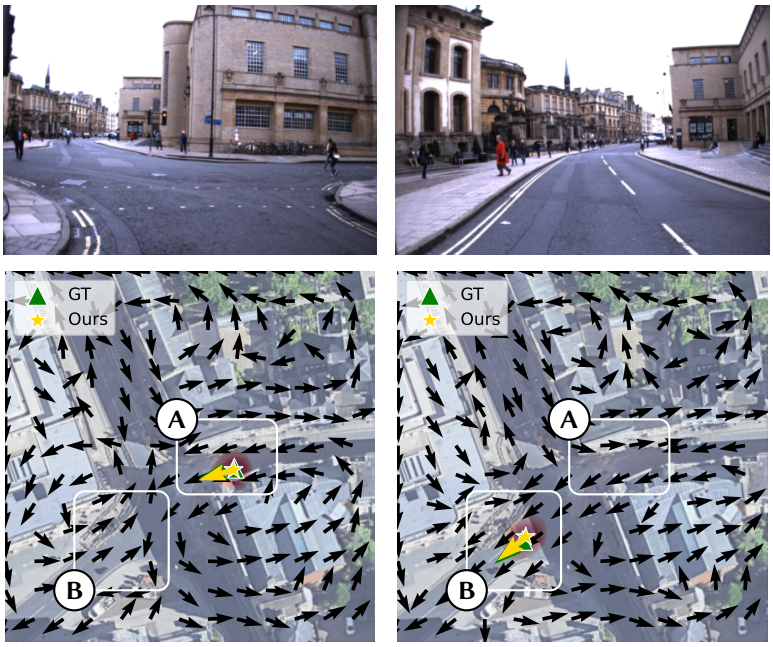


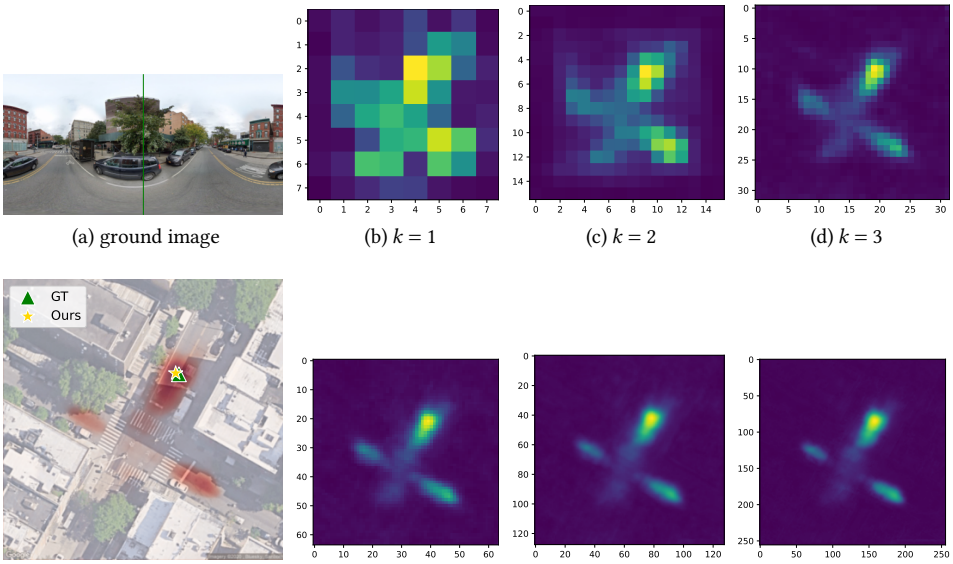
Figure 4.13: Localization and orientation estimation on two frames in a sequence on Oxford RobotCar Test 1 traversal. Right frame: ~13 seconds after the left frame. Because the two ground-level images capture a different scene, the output orientation field in the same region A (or B) in the identical aerial images is different.

lower levels can provide a better starting point for descriptor matching at higher levels, leading to better localization performance.

Qualitatively, it is seen in Figure 4.14 that the max matching score map inside the LMU module becomes sharper when the level k increases, but does not improve noticeably anymore after level 4, i.e. resolution 64×64, on the selected example. Quantitatively, the increase in localization performance is also less when k increases. For the main experiments, the proposed model includes LMU modules at all levels, i.e. $K = 6$. This setting also aligns with the commonly used coarse-to-fine formulation in other computer vision tasks [151, 152].

Other architectural variations: This section first compares the proposed model to [31]. Then, it studies the effect of the backbone, the OMU module, the number of orientation channels R, and the L2-normalization before feature concatenation in the LMU and OMU modules. Finally, this section tests replacing the Rolling & Matching with a simple concatenation of ground and aerial features, as well as removing the contrastive learning loss $\mathcal{L}_{\mathcal{M}}$.

In our conference work [31], VGG [159] and SAFA modules [106] are used as feature extractors and feature projectors, and the ground-to-aerial matching is only conducted at the bottleneck. Thus, for a fair comparison, a CCVPE model with VGG [159] backbone is included and it uses the LMU module only at the bottleneck, i.e. '$K = 1$, VGG'. In [31], pose estimation is achieved by comparing the model's probability estimations on differently



(a) ground image (b) $k = 1$ (c) $k = 2$ (d) $k = 3$

(e) localization on aerial image (f) $k = 4$ (g) $k = 5$ (h) $k = 6$

Figure 4.14: Visualization of the max matching score map in LMU at levels 1 to 6.

rotated samples, while this chapter constructs orientation-aware descriptors and estimates the pose in a single forward pass. Validation results of models with different settings are summarized in Table 4.4. Our model with VGG [159] backbone and $K = 1$ beats the approach in [31] in both localization and orientation accuracy.

The comparison between the row '$K = 1$, VGG' and the row '$K = 1$' in Table 4.4, shows that using EfficientNet-B0 [157] as the default backbone improves both localization and orientation estimation. Interestingly, it is found that if the aerial descriptors are not normalized before the feature concatenation in the LMU and OMU modules, both localization and orientation estimation performance decrease significantly. In particular, the orientation estimation becomes no better than a random guess. The magnitude of cosine similarity matching score $\mathcal{M}_k$ is between -1 and 1. Normalizing the aerial descriptors makes the magnitude of their elements stay in a similar range as $\mathcal{M}_k$. If the concatenated aerial descriptors are not normalized, the model might not effectively use the information in matching score $\mathcal{M}_k$ and treat M_k as noise.

Similar to concatenating aerial descriptors without normalization, excluding the OMU module and only processing the aerial descriptors also makes the orientation prediction fail. Since the ground images' orientation is randomly changed during training, there is no useful prior on the orientation when only considering the aerial image. Next, this section studies the effect of different numbers of orientation channels R. When increasing the granularity in the orientation space, i.e. using a larger R when rolling aerial descriptors in the LMU and OMU modules, both localization and orientation estimation performance increases. Constructing aerial descriptors for more global orientation intervals not only provides more fine-grained orientation matching scores but also improves the orientation-

aware features for localization. Limited by the width W' of encoded ground feature $g_e(G)$, the maximum R the proposed model can use is 20 on the VIGOR dataset. It is observed a small increase in mean orientation error when increasing R from 10 to 20. Overall, $R = 20$ provides the best localization result, and therefore it is used in the main experiments ($R = 16$ is used on KITTI because the input image has a different resolution). Similar to the Localization Decoder, this section tested including 6 OMU modules for all 6 levels in the Orientation Decoder. Although this setting reduces the median orientation error, there is an increase in the localization error and mean orientation error. LMU at higher levels has finer spatial resolutions, while the granularity in orientation space is fixed, e.g. $R = 20$, in all OMU modules. Thus, the same benefit is not expected here as in the Localization Decoder, and the OMU module is only used at the first level.

When replacing the proposed Rolling & Matching by straightforward ground-aerial feature concatenation, there is a large drop in both localization and orientation estimation performance, see the comparison between '$K = 1$' and 'Concat@1' and the comparison between '$K = 6, R = 20$' and 'Concat@6' in Table 4.4. When directly concatenating the ground feature with the aerial feature, the model has the additional challenge of learning that different rotated versions of the same panoramic image should be located at the same place. In contrast, our Rolling & Matching design injects inductive biases into the model by using the translational equivariant ground encoder, and by forcing corresponding ground and aerial descriptors to be similar. Specifically, the model ensures this rotational equivariance is kept by the OMU for orientation estimation. Therefore, when inputting different rotated versions of the same panorama, the same matching score pattern would re-occur in different orientation channels. Our Orientation Decoder still needs to learn how different permutations of matching scores translate to the orientation vector field, but this matching volume has a relatively low number of channels compared to concatenated ground and aerial features. The LMU's inductive bias is to be *invariant* to different ground camera's orientations, which is achieved by taking the maximum over orientation channels. None of these orientation and localization-specific inductive biases are present in the concatenation approach, which explains the large difference in performance.

Importantly, our Rolling & Matching is empowered by the orientation-aware ground and aerial descriptors. If the orientation awareness for the aerial descriptor is not enforced, i.e. by removing the infoNCE losses, both localization and orientation prediction performance of the model decreases significantly, see rows with 'No infoNCE' in Table 4.4.

Loss on localization heat map: Using the best model architecture, here compares the cross-entropy loss and Wasserstein distance-based loss for localization and uncertainty estimation. Table 4.5 shows the mean and median localization error and the predicted probability at the ground truth pixel of models trained with different losses.

The model trained with cross-entropy loss has a lower mean localization error than the model trained with Wasserstein distance-based loss. Notably, the model trained with Wasserstein distance-based loss outputs localization distributions that are very sharp. Biased by a few accurate predictions, the mean probability at the ground truth pixel of this model is higher than that of the model trained with cross-entropy. However, the median probability at the ground truth pixel is near zero, indicating many of the ground truth locations receive little probability mass. In temporal filtering or multi-sensor fusion, fusing such predictions might make the system miss the ground truth location. Besides, it is also

Table 4.4: CCVPE architecture comparisons on VIGOR Same-Area validation set. **Best in bold.** 'No norm' means the aerial descriptors are not normalized before feature concatenation in the LMU and OMU modules. '6 OMUs' means the OMU module at all levels is included.'Concat' denotes direct concatenating of the ground and aerial features instead of conducting Rolling & Matching.

VIGOR, val.	↓ Localization (m)		↓ Orientation (°)	
	mean	median	mean	median
[31]	9.76	6.15	55.91	15.66
K=1, VGG	7.06	3.90	18.12	7.91
K=1, No norm	5.44	2.95	89.72	89.17
K=1, No infoNCE	9.37	6.04	90.59	90.53
K=1	4.93	2.61	14.68	7.50
Concat@1	13.57	11.87	89.29	89.17
K=6, No OMU	3.73	**1.40**	89.92	89.86
K=6, No infoNCE	9.11	5.67	90.22	90.17
K=6, R=2	5.35	2.40	36.74	25.39
K=6, R=4	4.51	1.80	16.06	8.57
K=6, R=5	4.21	1.74	15.11	8.42
K=6, R=10	3.85	1.51	**13.06**	6.88
K=6, R=20 (default)	**3.63**	1.42	13.11	6.61
K=6, 6 OMUs	3.78	1.43	15.34	**3.81**
Concat@6	9.66	6.63	89.60	89.03

Table 4.5: Evaluation of CCVPE with different localization losses on VIGOR Same-Area validation set. **Best in bold.**

VIGOR, val.	↓ Localization (m)		↑ P@GT (1×10^{-3})	
	mean	median	mean	median
Cross-entropy	**3.63**	1.42	1.48	**1.00**
Wasserstein	3.75	**1.41**	**3.97**	0.00

observed that training with Wasserstein distance-based loss makes the output distribution less indicative of localization and orientation errors. This reduces its practicality in safety crucial applications where the outliers in prediction should be filtered out. Thus, the cross-entropy loss is used as the localization loss $\mathcal{L}_D$.

4.3.10 Runtime analysis

First, this section studies how the proposed Rolling & Matching influences the runtime of our method. On the VIGOR dataset, when increasing the number of orientation bins ($R = 2, 4, 5, 10, 20$) for Rolling & Matching in all LMU and OMU modules, the inference speed of our method decreases slightly (18, 17, 17, 17, 15 FPS). Since the Rolling & Matching is a convolution process between ground and aerial descriptors, it can be done efficiently.

Next, this section compares the runtime of our method to that of previous state-of-the-art methods. To include more baselines, the comparison is done on the KITTI dataset. On the same device (a single V100 GPU), our method takes 0.042s to process a pair of input

images (24 FPS) on the KITTI dataset, which is slower than SliceMatch's 156 FPS [33] but faster than LM's 0.59 FPS [130]. Importantly, even though SliceMatch runs faster, CCVPE is considerably more accurate in localization. Note that the authors of [134] evaluated the runtime of their method on the KITTI-360 dataset with a more advanced GPU (RTX6000), and their method runs at approximately 2-3 Hz [134], which is slower than CCVPE.

4.3.11 Perspective or equirectangular projected images

As discussed in Section 4.2.3, the proposed CCVPE model assumes that ground images follow a cylindrical projection, namely that each column of pixels in the image represents the same number of degrees in the horizontal FoV. In practice, regular (non-panoramic) camera images with a small horizontal FoV, e.g. FoV $< 90^{\circ}$, are rectified using a perspective projection rather than a cylindrical one. In Section 4.2.3, it is claimed that using perspective images as input to our CCVPE model still works well, even though it somewhat violates the cylindrical assumption. In this section, detailed experimental results are provided to support the claim.

This section compares training and testing the proposed CCVPE model using the regular perspective images in KITTI [20] and Oxford RobotCar datasets [1] to using the same image after converting them from perspective to cylindrical, or more specifically, equirectangular, projected images. As the detailed results for both datasets in Table 4.6 and 4.7 show, it can be observed that there is little performance difference between models using the different types of projections. Thus, it is proposed to directly input perspective images into the model instead of including extra image pre-processing.

Note that the ground encoder is trained end-to-end, hence the ground descriptors can be optimized for robustness against slight violations of the assumption that a certain number of horizontal pixels correspond to a certain number of degrees in the horizontal FoV. Plus, when the horizontal FoV of the perspective image is relatively small, as is the case for images in the KITTI [20] and Oxford RobotCar [1] datasets, the difference between the perspective image and the equirectangular projected images is also small, see an example from Oxford RobotCar in Figure 4.15.

Figure 4.15: An example ground image from Oxford RobotCar dataset. Left: perspective projection (no image processing was applied). Right: equirectangular projection (pre-processed image).

Table 4.6: Comparison between using original perspective images (no image processing was applied) and equirectangular projected images (pre-processed images) for training and testing on the KITTI dataset. **Best in bold.** In the leftmost column, ±10° denotes training and testing with an orientation prior with noise in the range [−10°, 10°], and 360° means no orientation prior by using noise from the 360° circular domain.

Same-Area		↓ Loc. (m)		↑ Lateral (%)			↑ Longitudinal (%)			↓ Ori. (°)		↑ Ori. (%)		
		mean	med.	1m	3m	5m	1m	3m	5m	mean	med.	1°	3°	5°
±10°	Perspective	1.22	**0.62**	97.35	98.65	99.71	77.13	96.08	97.16	**0.67**	**0.54**	**77.39**	**99.47**	**99.95**
	Equirectangular	**1.11**	**0.62**	**97.67**	**98.67**	**99.84**	**81.13**	**96.34**	**97.51**	0.69	0.56	76.23	99.44	99.92
360°	Perspective	6.88	3.47	53.30	77.63	85.13	25.84	55.05	68.49	15.01	6.12	8.96	26.48	42.75
	Equirectangular	**6.79**	**3.33**	**55.00**	**78.74**	**85.98**	**26.85**	**55.63**	**68.70**	**13.64**	**5.62**	**9.73**	**29.47**	**45.16**
Cross-Area		↓ Loc. (m)		↑ Lateral (%)			↑ Longitudinal (%)			↓ Ori. (°)		↑ Ori. (%)		
		mean	med.	1m	3m	5m	1m	3m	5m	mean	med.	1°	3°	5°
±10°	Perspective	**9.16**	**3.33**	**44.06**	**81.72**	**90.23**	23.08	**52.85**	**64.31**	**1.55**	**0.84**	**57.72**	**92.34**	**96.19**
	Equirectangular	10.91	3.35	43.79	79.85	87.30	**24.26**	52.13	63.30	2.78	0.92	53.54	89.02	92.84
360°	Perspective	**13.94**	10.98	23.42	49.15	60.46	11.81	**29.85**	**42.12**	77.84	**63.84**	**3.14**	**9.24**	**14.56**
	Equirectangular	13.96	**10.93**	**23.89**	**50.28**	**61.81**	**12.00**	29.75	41.81	**77.73**	65.44	2.72	8.29	13.94

Table 4.7: Comparison between using original perspective images (no image processing was applied) and equirectangular projected images (pre-processed images) for training and testing on the Oxford RobotCar dataset. **Best in bold.**

Projection	Test set 1				Test set 2				Test set 3			
	↓ Loc. (m)		↓ Ori. (°)		↓ Loc. (m)		↓ Ori. (°)		↓ Loc. (m)		↓ Ori. (°)	
	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median
Perspective	1.24	1.02	**1.70**	**1.32**	**1.56**	**1.16**	**1.91**	**1.10**	**1.55**	**0.92**	**2.50**	**1.13**
Equirectangular	**1.23**	**0.97**	2.74	2.06	1.63	1.19	2.88	2.07	1.65	1.10	3.05	1.85

4.4 Conclusion of the chapter

In this chapter, the novel Convolutional Cross-View Pose Estimation method (CCVPE) was proposed. CCVPE exploits the strength of a translational equivariant feature encoder and of contrastive learning to learn orientation-aware descriptors for joint localization and orientation estimation. Instead of estimating a single location, its Localization Decoder outputs a multi-modal distribution to capture the underlying localization uncertainty. The Localization Matching Upsampling (LMU) and Orientation Matching Upsampling (OMU) modules were devised to summarize orientation invariant localization cues and orientation-dependent information from the descriptor matching result when upsampling the aerial feature maps inside two separate decoders. The Orientation Decoder outputs a dense orientation vector field that is conditioned on the localization distribution. Thus, CCVPE's orientation prediction becomes multi-modal when there are multiple modes in the localization distribution.

CCVPE achieves 72% and 36% lower median localization errors (1.42 m and 3.47 m) than the previous SOTA (5.07 m and 5.41 m) on the VIGOR and KITTI datasets, and it has comparable orientation estimation accuracy. Importantly, CCVPE can work with ground images with different horizontal FoVs and incorporate an orientation prior to improve the localization without re-training. Its probabilistic output can be used to filter out predictions that potentially have large localization and orientation errors, yielding better practicality than the baselines that do not have a probability estimate. It is demonstrated on traversals collected at different times in the Oxford RobotCar dataset that CCVPE can estimate the pose of ego-vehicle at 14 FPS with a median lateral and longitudinal error below 1 meter and a median orientation error around 1°, bringing cross-view pose estimation methods closer

to the requirement of autonomous driving of < 0.3 m lateral and longitudinal localization accuracy.

5

Cross-view camera pose estimation by geometry-guided feature aggregation

This chapter is based on *T. Lentsch*, Z. Xia*, H. Caesar, and J. F. P. Kooij, "SliceMatch: Geometry-guided Aggregation for Cross-View Pose Estimation," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17225–17234, 2023* [33], *:equal contribution.

5.1 Overview

Recently, several works have addressed cross-view camera localization [127] or 3-DoF pose estimation [31, 110, 130, 135]. Roughly, those methods can be categorized into global image descriptor-based [31, 127] and dense pixel-level feature-based [110, 130, 135] methods. Global descriptor-based methods take advantage of the compactness of the image representation and often have relatively fast inference time [31, 127]. Dense pixel-level feature-based methods [110, 130, 135] are potentially more accurate as they preserve more details in the image representation. They use the geometric relationship between the ground and aerial view to project features across views and estimate the camera pose via computationally expensive iterations. Aiming for both accurate and efficient camera pose estimation, this dissertation improves the global descriptor-based approach and enforces feature locality in the descriptor.

This dissertation observes several limitations in existing global descriptor-based cross-view camera pose estimation methods [31, 127]. First, they rely on the aerial encoder to encode all spatial context and the aerial encoder has to learn how to aggregate local information, e.g., via the SAFA module [106], into the global descriptor, without accessing the information in the ground view or exploiting geometric constraints between the ground-camera viewing frustum and the aerial image. Second, existing global descriptor-based methods for cross-view localization [31, 127] do not explicitly consider the orientation of the ground camera in their descriptor construction. As a result, they either do not estimate the orientation [127] or require multiple forward passes on different rotated samples to infer the orientation [31]. Third, existing global descriptors-based methods [31, 127] are not trained discriminatively against different orientations. Therefore, the learned features may be less discriminative for orientation prediction.

To address the observed gaps, this chapter devises a novel, accurate, and efficient method for cross-view camera pose estimation called SliceMatch (see Figure 5.1). Its novel aerial feature aggregation explicitly encodes directional information and pools features using known camera geometry to aggregate the extracted aerial features into an aerial global descriptor. The proposed aggregation step 'slices' the ground Horizontal Field-of-View (HFoV) into orientation-specific descriptors. For each pose in a set of candidates, it aggregates the extracted aerial features into corresponding aerial slice descriptors. The aggregation uses cross-view attention to weigh aerial features w.r.t. to the ground descriptor, and exploits the geometric constraint that every vertical slice in the ground image corresponds to an azimuth range extruding from the projected ground camera position in the aerial image. The feature extraction is done only once for constructing the descriptors for all pose candidates, resulting in fast training and inference speed. The model is trained contrastively by pairing the ground image descriptor with aerial descriptors at different locations and orientations. Hence, the model learns to extract discriminative features for both localization and orientation estimation.

The main contributions of this chapter include: i) A novel aerial feature aggregation step that uses a cross-view attention module for ground-view guided aerial feature selection, and the geometric relationship between the ground camera's viewing frustum and the aerial image to construct pose-dependent aerial descriptors. ii) SliceMatch's design allows for efficient implementation, which runs significantly faster than previous state-of-the-art methods. Namely, for an input ground-aerial image pair, SliceMatch extracts dense features

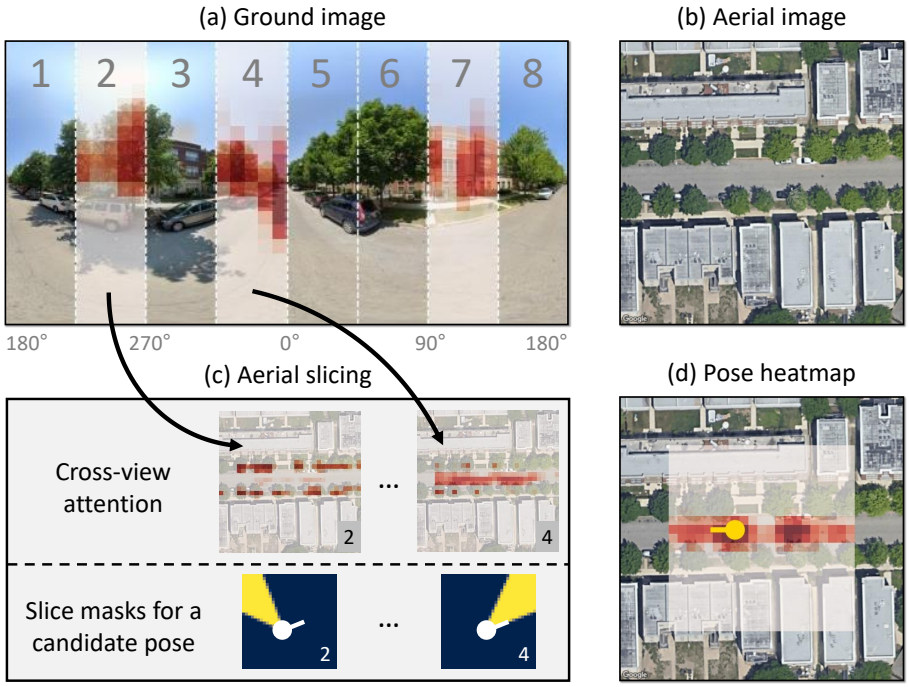


Figure 5.1: **SliceMatch identifies for a ground-level image (a) its camera's 3-DoF pose within a corresponding aerial image (b).** It divides the camera's Horizontal Field-of-View (HFoV) into 'slices', i.e., vertical regions in (a). After self-attention, our novel aggregation step (c) applies cross-view attention to create ground slice-specific aerial feature maps. To efficiently test many candidate poses, the slice features are aggregated using pose-dependent aerial slice masks that represent the camera's sliced HFoV at that pose. The slice masks for each pose are precomputed. All aerial pose descriptors are compared to the ground descriptor, resulting in a dense scoring map (d). Our output is the best-scoring pose.

only once, aggregates aerial descriptors at a set of poses without extra computation, and compares the aerial descriptor of each pose with the ground descriptor. iii) Compared to the previous state-of-the-art global descriptor-based cross-view camera pose estimation method, SliceMatch constructs orientation-aware descriptors and adopts contrastive learning for both locations and orientations. Powered by the above designs, SliceMatch sets the new state-of-the-art for cross-view pose estimation on two commonly used benchmarks.

5.2 Methodology

This section explains the cross-view camera pose estimation task, the proposed SliceMatch method, and its novel aggregation step.

5.2.1 Cross-view camera pose estimation

Given a ground-level image G and a square overhead aerial image A that contains the local surroundings of G, this chapter aims to determine the 3-DoF pose, $\xi = (u, v, \theta)$, of the ground camera that captured G. Here, $(u, v) \in [0, 1]^2$ are the image coordinates in A, and

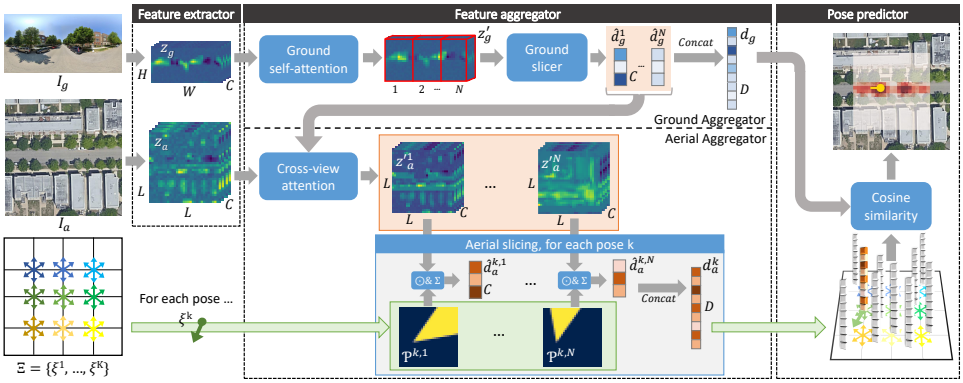


Figure 5.2: **The SliceMatch pipeline.** The input to SliceMatch is a pair of ground-aerial images and a set of K candidate ground camera poses. SliceMatch consists of ground and aerial feature extractors, feature aggregators, and a pose predictor. In the shown output image, the matching scores for all poses are overlayed on the input aerial image. The predicted pose is the one with the highest score.

5

$\theta \in [0, 360^\circ)$ is the camera orientation, i.e., the angle from the North direction clockwise to the center line (the 'front' direction) of the ground camera projected onto the aerial view. Ground images can either be panoramic or have a limited HFoV. Similar to [130], it is assumed that the ground camera's pitch and roll are small.

5.2.2 SliceMatch overview

SliceMatch explicitly separates feature extraction and aggregation, where the latter exploits geometric knowledge on how the ground camera's viewing frustum projects on the aerial image. In SliceMatch, pose estimation is formulated as an efficient process that compares aerial descriptors for a set $\Xi = \{\xi^1, \cdots, \xi^K\}$ of K candidate poses to the ground image descriptor. During training, the set consists of K_{train} poses at a fixed uniform grid in 3-DoF pose space. During inference, SliceMatch uses K_{test} poses ($K_{test} > K_{train}$), and the predicted pose is the candidate for which its aerial descriptor is most similar to the ground descriptor. See Figure 5.2 for an overview of the method. The next paragraphs discuss each step.

Feature extractor: Input images G and A are first mapped to feature maps, $z_g = f_g(G) \in \mathbb{R}^{H \times W \times C}$ and $z_a = f_a(A) \in \mathbb{R}^{L \times L \times C}$, where f_g and f_a can be any convolutional backbone (e.g. VGG [159] or ResNet [160]). SliceMatch adopts the commonly used setup that f_g and f_a have the same architecture without weight-sharing [31, 127]. It seeks translational equivariance in its encoders and thus does not focus on Vision Transformers [133].

Feature aggregator: The proposed novel aggregator step efficiently constructs a single ground and multiple pose-dependent aerial descriptors from the extracted image features through the use of 'slices'. In this chapter, each slice represents a non-overlapping range in the azimuth viewing direction, and is used to aggregate the local image features within that azimuth range. In the ground view, a slice thus corresponds to a vertical rectangular region in the image/feature map, and in the aerial view, it is a triangle-shaped region extending from a candidate pose (see Figure 5.1). This will be explained in more detail in Section 5.2.3. This chapter refers to an aggregated feature in a single slice as a ground/aerial

slice descriptor, containing the visual information for that viewing direction. Likewise, this chapter refers to a ground/aerial *global descriptor* as the concatenation of the slice descriptors of all pose-relative orientations, representing the full HFoV of the ground camera.

Concretely, the extracted feature maps z_g and z_a are fed into the proposed heterogeneous feature aggregators, as shown in Figure 5.2. The ground aggregator $agg_g(z_g)$ generates a set[1] $\widehat{\Delta}_g = \{\hat{d}_g^1, \cdots, \hat{d}_g^N\}$ of C-dimensional slice descriptors $\hat{d}_g^n$ for N azimuth directions, where N is a hyperparameter for the number of slices. The ground global descriptor $d_g = Concat(\hat{d}_g^1, \cdots, \hat{d}_g^N)$ is thus a vector of length $D = N \cdot C$. The aerial aggregator $agg_a(z_a, \widehat{\Delta}_g, \Xi)$ receives the aerial features z_a, the ground slice descriptors $\widehat{\Delta}_g$, and the set of K poses Ξ. It generates $\Delta_a = \{d_a^1, \cdots, d_a^K\}$, the set of K pose-dependent aerial global descriptors $d_a^k \in \mathbb{R}^D$. Section 5.2.3 will discuss both aggregators in detail.

Pose predictor: The pose predictor receives the ground global descriptor d_g, and the set Δ_a that contains the K aerial global descriptors corresponding to the candidate poses in set Ξ. SliceMatch computes the cosine similarity c^k between d_g and all $d_a^k \in \Delta_a$ and, during inference, uses ξ^k corresponding to the highest similarity value $c_{max}^k = max(c^1, \cdots, c^K)$ as the predicted pose. Note that similar to [31], SliceMatch obtains a heatmap that can express multimodal pose estimation ambiguity, which can be beneficial for downstream fusion.

Loss Function: This chapter modifies the infoNCE loss [154] from contrastive representation learning [161] to train SliceMatch. Using $K = K_{train}$ training poses, our loss $\mathcal{L}$ is defined as,

$$\mathcal{L} = -\log\left(\frac{\exp(c^{GT}/\tau)}{\frac{\alpha}{K}\sum_{k=1}^{K}\exp(c^k/\tau) + \exp(c^{GT}/\tau)}\right). \tag{5.1}$$

In Equation (5.1), α is the introduced hyperparameter that weighs the contribution of K poses to the learning. Variable c^{GT} is the cosine similarity between d_g and d_a^{GT} at ξ^{GT}, and c^k is that between d_g and d_a^k at ξ^k. Hyperparameter τ is proposed in [154]. The original infoNCE loss in [154] can be acquired using $\alpha = K$. With $\mathcal{L}$, the ground truth pose is contrasted with K_{train} other poses at different locations and orientations, thus the model learns to extract discriminative features for both location and orientation prediction.

5.2.3 Geometry-guided cross-view aggregation

Here, this section describes the novel aggregation step in more detail. Unlike the SAFA module [106] used in [31, 127], the proposed aggregation uses geometric knowledge on how the views should spatially relate. Ground-to-aerial attention further improves quality, as the visual information in each ground slice informs what aerial features are relevant to produce the corresponding aerial slice descriptors, thus promoting shared features specific to each viewing direction.

Ground Feature Aggregator:

To summarize the important features in each vertical slice in the ground camera's viewing frustum, SliceMatch constructs its ground feature aggregator $agg_g(z_g)$ with a self-attention module and a feature slicer. Since not all information in ground image G will

[1] Note that $\hat{d}$ and $\hat{\Delta}$ (with hat) are used to indicate *slice* descriptors/sets, and d and Δ (without hat) are used to indicate *global* descriptors/sets.

be present in the aerial image A (e.g. sky and transient objects), the self-attention module re-weighs z_g along the spatial dimensions H and W,

$$z'_g = \mathcal{M}_g \odot z_g, \quad \mathcal{M}_g = Sigmoid(Conv_{1\times1}(z_g)). \tag{5.2}$$

Here, $\mathcal{M}_g$ is a learned mask with shape $H \times W \times 1$ that re-weighs the ground feature map z_g into z'_g. The *Sigmoid* operation enforces the weights in $\mathcal{M}_g$ are between 0 and 1. The $\odot$ denotes element-wise multiplication, with the ability to broadcast the mask $\mathcal{M}_g$ over all channels of z_g.

The ground slicer then divides z'_g into N vertical slices, cutting the feature map along the horizontal (azimuth) direction. For each slice, a normalized slice descriptor is computed by averaging all features within the slice and applying L2 normalization. This results in the set $\widehat{\Delta}_g = \{\hat{d}_g^1, \cdots, \hat{d}_g^N\}$ of N ground slice descriptors. Each slice local descriptor thus represents the model's attended feature in the corresponding vertical slice (i.e. an azimuth range) in the ground camera's viewing frustum. The ground global descriptor is obtained by concatenating all N ground slice descriptors, i.e. $d_g = Concat(\hat{d}_g^1, \cdots, \hat{d}_g^N)$.

Aerial Feature Aggregator:

The aerial aggregator $agg_a(z_a, \widehat{\Delta}_g, \Xi)$ has a similar role as the ground aggregator, but its feature selection is also conditioned on the ground slice descriptors $\widehat{\Delta}_g$ using a cross-view attention module and the set of poses Ξ for geometry-guided feature aggregation.

Cross-view attention: Since in the ground view most content that is seen in the aerial view will be occluded, SliceMatch proposes a cross-view attention module to specifically extract the aerial features that should match the visible content of each ground slice. In detail, SliceMatch matches the C-dimensional aerial feature $z_a^{i,j}$ at each spatial location (i, j) with $1 \leq i \leq L, 1 \leq j \leq L$ in the aerial feature map z_a to each ground slice descriptor $\hat{d}_g^n \in \widehat{\Delta}_g$ to acquire a similarity score map S^n of size $L \times L$, where $S^{n,i,j} = Sim(\hat{d}_g^n, z_a^{i,j})$. In total, there are N similarity score maps, i.e. one for each ground slice descriptor. Then, SliceMatch treats each S^n as extra features and concatenates it along the feature dimension with aerial feature map z_a [31], and uses these extended features to produce a cross-view attention mask,

$$\mathcal{M}_a^n = Sigmoid(Conv_{1\times1}(Concat(z_a, S^n))). \tag{5.3}$$

Thus, there are in total N cross-view masks $\mathcal{M}_a^n$. Each of these denotes the importance of the aerial features w.r.t. the n-th ground slice descriptor $\hat{d}_g^n$. Finally, SliceMatch re-weighs z_a for each ground slice descriptor, giving us N re-weighted aerial feature maps $z_a'^n$ of size $L \times L \times C$, i.e. $z_a'^n = \mathcal{M}_a^n \odot z_a$.

Geometry-guided feature aggregation: Finally, the K pose-dependent aerial descriptors d_a^k can be constructed for the candidate poses in Ξ. For each pose ξ^k, N aerial slice masks $\mathcal{P}^{k,n} \in [0,1]^{L\times L}$, $1 \leq n \leq N$, can be precomputed. The slice mask $\mathcal{P}^{k,n}$ expresses the geometry of the ground camera's viewing frustum in the aerial feature map for the n-th orientation slice, assuming that the camera would have the k-th pose. Each cell in the slice mask contains a value in the range $[0,1]$ proportional to how much of that cell intersects this frustum, so 1.0 for fully contained cells, 0.0 for cells fully outside the frustum, and an intermediate value for cells that partially overlap.

With the slice masks, the n-th aerial slice descriptor at pose k can be computed efficiently. For each of the C channels, SliceMatch computes a weighted average over all of the $L \times L$ spatial locations (i, j) in the feature map $z_a'^n$, using the elements of slice mask $\mathcal{P}^{k,n}$ as weights. After L2 normalization, It obtains aerial slice descriptor $\hat{d}_a^{k,n}$,

$$\hat{d}_a^{k,n} = Norm\Big(\frac{1}{\sum_{i,j}\mathcal{P}_{i,j}^{k,n}} \sum_{i,j} \big(\mathcal{P}^{k,n} \odot z_a'^n\big)_{i,j}\Big). \tag{5.4}$$

Analogous to the ground view, the k-th pose's global descriptor is obtained using $d_a^k = Concat(\hat{d}_a^{k,1}, \cdots, \hat{d}_a^{k,N})$.

Efficient implementation: A benefit of the proposed architecture is that the computational complexity of most operations is independent of the number of candidate poses K. The main cost to increase K, and therefore improve accuracy by testing more diverse poses at inference time, is to add more precomputed slice masks, and perform the additional multiplications and normalizations for Equation (5.4) and the final cosine similarity comparison. These are simple operations that can be highly optimized and parallelized in the implementation, and it will be shown that testing more candidate poses does not increase our runtime.

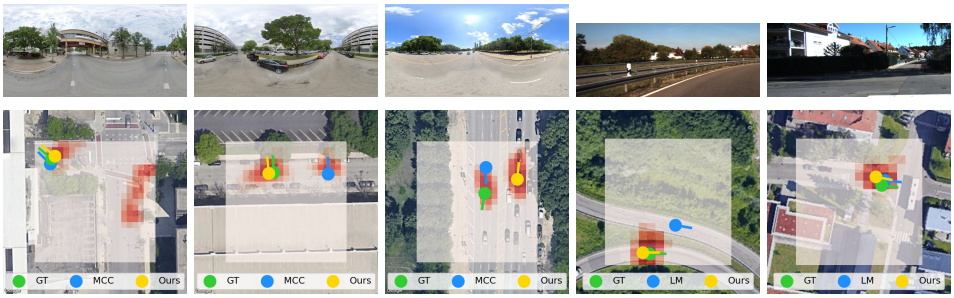
5.3 Experiments

This section first introduces the used datasets and the evaluation metrics. After that, the implementation details and ablation studies are presented. Finally, SliceMatch is quantitatively and qualitatively compared to state-of-the-art baselines.

5.3.1 Datasets

VIGOR dataset [127] contains geo-tagged ground-level panoramas and aerial images collected in 4 cities in the US. As defined in [127], each ground panorama has 1 positive and 3 semi-positive aerial images. An aerial image is positive if the ground camera's location is within the aerial image's center quarter area, otherwise, it is semi-positive. Importantly, this dissertation found that the original ground truth locations in [127] can contain errors up to 3 meters due to the use of wrong ground resolutions (0.114m/pixel) of the aerial images, thus this dissertation created and uses here corrected labels. For training and testing the proposed method and baselines, experiments use positive aerial images and corrected ground truth (this dissertation reran all baselines since quantitative results with the new labels differ slightly from those reported in the literature). Experiments adopt the same-area and cross-area splits from [127] to test the model's generalization to new measurements in the same cities and across different cities. Besides, the same-area training dataset of New York is used as a tuning split for the ablation study.

KITTI dataset [20] contains ground-level images with a limited HFoV taken by a moving vehicle from different trajectories at different times and [130] augmented the dataset with aerial images. This dissertation uses their split. The Training and Test1 sets are different measurements from the same region, while the Test2 set has been captured in a different region.



(a) VIGOR example 1 (b) VIGOR example 2 (c) VIGOR example 3 (d) KITTI example 1 (e) KITTI example 2

Figure 5.3: **Qualitative evaluation of SliceMatch on VIGOR [127] and KITTI [130, 162].** Top row: input ground image. Bottom row: GT and pose estimation results overlayed on input aerial image. Red shading indicates highest similarity score between the ground descriptor and the aerial descriptors among all orientations at that location. (c) shows a SliceMatch failure: the best match is in the wrong mode.

5

5.3.2 Evaluation metrics

This chapter follows the convention of [31] and reports the mean and median error in meters between the predicted and ground truth location over all test image pairs. Similarly, for orientation prediction, this chapter reports the mean and median absolute angular difference between the predicted and ground truth orientation in degrees. Following [130], for the KITTI dataset, this chapter additionally includes the recall under a certain threshold for longitudinal (driving direction) and lateral localization error, and orientation estimation error. The thresholds are set to 1m and 5m for localization and to 1° and 5° for orientation estimation.

5.3.3 Implementation details

As in [31, 127], the proposed SliceMatch uses VGG16 [159] up to stage 5 for the feature extractors f_g and f_a. The pooling operation of the last layer is removed. The spatial size of I_g is 320×640 on VIGOR dataset and 256×1024 on KITTI dataset, and that of I_a is 512×512 on both datasets. This results in feature maps with $H \times W = 20 \times 40$ / 16×64 on VIGOR / KITTI, $L \times L = 32 \times 32$, and $C = 512$. The feature extractors do not share their weights and are pre-trained on ImageNet [146]. In Equation (5.2) and (5.3), $Conv_{1\times1}$ consists of two sequential convolution layers with a kernel size of 1 and a ReLU activation in between. During training, SliceMatch is trained end-to-end using Adam optimizer [147] with a learning rate of 1×10^{-5}, and this chapter uses a batch size of 4. To get a set of candidate camera poses Ξ, the default setting of SliceMatch uses poses at a uniform grid of 7×7 locations $\times$ 16 orientations on VIGOR, and 5×5 locations $\times$ 16 orientations on KITTI during training. For inference, the default setting uses $21 \times 21 \times 64$ and $15 \times 15 \times 64$ poses, respectively. This results in $K_{train} = 784$ and $K_{test} = 28224$ on VIGOR, and $K_{train} = 400$ and $K_{test} = 14400$ on KITTI.

5.3.4 Baselines

Experiments compare SliceMatch to state-of-the-art global descriptor-based methods Cross-View Regression (CVR) [127] and Multi-Class Classification (MCC) [31] on the VIGOR

dataset[2]. Since CVR does localization with known orientation and, in [31], MCC mainly focuses on localization, this chapter compares SliceMatch to baselines for localization with known orientation and also for 3-DoF pose estimation. Following [31], this chapter trains CVR [127] for localization only (not retrieval) as it gives better localization results. On the KITTI dataset, SliceMatch is compared to dense local feature-based fine-grained image retrieval method DSM [121], and to iterative camera pose estimation method LM [130]. In [130], the LM method is trained and tested with a 20° prior on the ground camera's orientation. Experiments adopt the same setting and additionally provide the results with LM and SliceMatch trained and tested with unknown orientation. On both datasets, baselines are trained with inputs with the same size as used for SliceMatch.

5.3.5 Ablation study

Before other experiments, this section tests on the VIGOR tuning set using $\alpha \in \{2, 4, 8, 16, K\}$ for the loss of Equation (5.1), and tune the number of slices N. it is found that $\alpha = 4$ gives the best result, yielding 0.48m improvement on the mean localization error for our model compared to $\alpha = K$ in the original infoNCE loss [154]. If the number of slices N is small, the mean and median localization and orientation estimation errors increase (see Table 5.1). The model with $N = 1$ cannot infer orientation. When the width of the ground feature map W is not a multiple of N, SliceMatch interpolates the ground feature map z_g to acquire $\hat{d}_g^n$. However, it can be seen that the performance saturates above 16 slices. Next, this section tested SliceMatch without cross-view attention by dropping the concatenated S^n in Equation (5.3). Table 5.1 shows that including our proposed cross-view attention module brings a boost to both localization and orientation estimation performance. Thus, SliceMatch includes cross-view attention and uses $\alpha = 4$ and $N = 16$ in the following main experiments.

N	Cross-View Attention	↓ Location (m) Mean	↓ Location (m) Median	↓ Orientation (°) Mean	↓ Orientation (°) Median
1	X	12.73	11.51	-	-
4	✓	9.47	7.47	51.49	32.96
8	✓	9.16	6.81	37.68	15.58
16	✓	**7.60**	**5.23**	**29.27**	**9.22**
32	✓	8.14	5.31	32.01	10.31
16‡	✓	8.08	5.44	31.05	11.02
16	X	7.93	5.81	29.50	12.32

Table 5.1: **Location and orientation error for different slice number N values on the VIGOR tuning split.** ‡ indicates model trained with original infoNCE loss [154]. Best performance in **bold**.

Model	Backbone	Aligned Images	Same-Area ↓ Location (m) Mean	Same-Area ↓ Location (m) Median	Same-Area ↓ Orientation (°) Mean	Same-Area ↓ Orientation (°) Median	Cross-Area ↓ Location (m) Mean	Cross-Area ↓ Location (m) Median	Cross-Area ↓ Orientation (°) Mean	Cross-Area ↓ Orientation (°) Median
CVR [127]	VGG16	✓	8.99	7.81	-	-	8.89	7.73	-	-
MCC [31]	VGG16	✓	6.94	3.64	-	-	9.05	5.14	-	-
SliceMatch (ours)	VGG16	✓	**5.18**	**2.58**	-	-	**5.53**	**2.55**	-	-
MCC [31]	VGG16	X	9.87	6.25	56.86	16.02	12.66	9.55	72.13	29.97
SliceMatch (ours)	VGG16	X	**8.41**	**5.07**	**28.43**	**5.15**	**8.48**	**5.64**	**26.20**	**5.18**
SliceMatch (ours)	ResNet50	X	**6.49**	**3.13**	**25.46**	**4.71**	**7.22**	**3.31**	**25.97**	**4.51**

Table 5.2: **Location and orientation estimation errors on VIGOR [127].** *Aligned Images* means the ground image orientation is known. For *unaligned images*, the models estimate the 3-DoF ground camera pose. Best performance in **bold**.

Samearea	Prior	↓ Location (m) Mean	↓ Location (m) Med.	↑ Lateral (%) r@1m	↑ Lateral (%) r@5m	↑ Longitudinal (%) r@1m	↑ Longitudinal (%) r@5m	↓ Orientation (°) Mean	↓ Orientation (°) Med.	↑ Orientation (%) r@1°	↑ Orientation (%) r@5°
DSM [121]	20°	-	-	10.12	48.24	4.08	20.14	-	-	3.58	24.44
LM [130]	20°	12.08	11.42	35.54	80.36	5.22	26.13	**3.72**	**2.83**	**19.64**	**71.72**
SliceMatch (ours)	20°	**7.96**	**4.39**	**49.09**	**98.52**	**15.19**	**57.35**	4.12	3.65	13.41	64.17
LM [130]	X	15.51	15.97	5.17	25.44	4.66	25.39	89.91	90.75	0.61	2.89
SliceMatch (ours)	X	**9.39**	**5.41**	**39.73**	**87.92**	**13.63**	**49.22**	**8.71**	**4.42**	**11.35**	**55.82**

Table 5.3: **Location and orientation estimation error and recall on KITTI [130, 162].** *Prior* means the orientation is known with a certain amount of noise. *Long.* and *Orien.* are abbreviations for *Longitudinal* and *Orientation*, respectively. Best performance in **bold**. The results for DSM [121] are taken from [130] and the trained LM model provided by [130] is used for its evaluation.

5.3.6 Same-area generalization

Experiments test model generalization to new panoramic and limited HFoV ground images within the same area on VIGOR and KITTI. As shown in Table 5.2 Same-Area, SliceMatch surpasses CVR [127] and MCC [31] in terms of both localization with known orientation and 3-DoF camera pose estimation. Compared to MCC, in which location-wise discriminative features are learned, SliceMatch contrasts the learned global descriptors with aerial descriptors at different locations and orientations. Hence, it is more discriminative especially w.r.t. orientations, and has a 19% and 68% reduction in the median localization and median orientation error when the orientation of test ground images is unknown. This chapter uses VGG16 as our main backbone for a fair comparison to the baselines, though this chapter notes that SliceMatch's localization and orientation error decreases even further when using ResNet50 as backbone. Figure 5.3b shows that SliceMatch can express its multimodal uncertainty when the observed scene has a symmetric layout. However, it sometimes picks candidate poses at a wrong mode, resulting in large errors (see Figure 5.3c). Over all test samples, SliceMatch has a substantially lower median error than its mean

[2]This chapter re-trained and evaluated the existing baselines on our corrected ground truth locations (see Section 5.3.1). The improved ground truth and code for our model are available at `https://github.com/tudelft-iv/SliceMatch`.

Crossarea	Prior	↓ Location (m) Mean	Med.	↑ Lateral (%) r@1m	r@5m	↑ Longitudinal (%) r@1m	r@5m	↓ Orientation (°) Mean	Med.	↑ Orientation (%) r@1°	r@5°
DSM [121]	20°	-	-	10.77	48.24	3.87	19.50	-	-	3.53	23.95
LM [130]	20°	**12.58**	12.11	27.82	72.89	5.75	26.48	**3.95**	**3.03**	18.42	**71.00**
SliceMatch (ours)	20°	13.50	**9.77**	**32.43**	**86.44**	**8.30**	**35.57**	4.20	6.61	**46.82**	46.82
LM [130]	X	15.50	16.02	5.60	25.60	5.64	25.76	89.84	89.85	0.60	2.65
SliceMatch (ours)	X	**14.85**	**11.85**	**24.00**	**72.89**	**7.17**	**33.12**	**23.64**	**7.96**	**31.69**	**31.69**

Table 5.4: **Location and orientation estimation error and recall on KITTI [130, 162].** *Prior* means the orientation is known with a certain amount of noise. *Long.* and *Orien.* are abbreviations for *Longitudinal* and *Orientation*, respectively. Best performance in **bold**. The results for DSM [121] are taken from [130] and the trained LM model provided by [130] is used for its evaluation.

error for both localization and orientation estimation, indicating that the mean is skewed by such outliers. In practice, SliceMatch's multimodal uncertainty could be resolved by applying downstream a probabilistic temporal filter on its output [31].

As shown in Table 5.3, on the KITTI dataset, both camera pose estimation methods, LM [130] and SliceMatch, surpass the fine-grained image retrieval-based method DSM [121]. When the orientation prior is present, SliceMatch has 34% and 62% lower mean and median localization error than LM [130], and its recall@1m and recall@5m is higher than that of LM [130] for localization in both lateral and longitudinal directions. Notably, since the ground images in KITTI view in the driving direction with a limited HFoV, finding the location along the longitudinal direction is more challenging than that for the lateral direction. Thus, recall for longitudinal direction is considerably lower than that for lateral direction, and this trend applies to all compared methods. The iterative refinement LM method [130] shows its advantage in orientation prediction when the strong orientation prior is present. This chapter highlights that SliceMatch can work without this prior. In contrast, LM [130] relies on the projection of dense local features from the aerial view to ground view [130] and does not work when there is no same scene captured in the projected view and the ground view (see Table 5.3).

5.3.7 Cross-area generalization

Generalization to new ground images in different areas is a more difficult task than that in the same area since the test area can look very different from the training area (e.g. different cities in the VIGOR dataset). As shown in Table 5.2 Cross-Area, SliceMatch generalizes well under this challenging setting in terms of both localization and orientation estimation, while there observes more degeneration in the cross-area test performance of MCC [31]. MCC's feature decoder receives the full scene information from its encoder, while SliceMatch divides the observed scene into slices and seeks per-slice discriminative features, resulting in more robustness against the change of the scene. Again, using a ResNet50 backbone further improves our results.

On KITTI Test2 set (Table 5.4), SliceMatch achieves a lower median localization error than LM [130] when the 20° orientation prior is present in both training and testing. But our mean error is higher than LM [130] by 0.92m and LM [130] surpasses SliceMatch in orientation prediction when a strong prior is available. SliceMatch performs considerably

better when no orientation prior is available as LM [130] gets stuck in local optima.

5.3.8 Runtime analysis

This section compares the runtime of SliceMatch to that of baselines on the same hardware, a single NVIDIA Tesla V100 GPU. For all baselines, the released code from their authors is used. CVR [127] and MCC [31] are implemented in TensorFlow, LM [130] and our SliceMatch in PyTorch. The frames per second (FPS) are calculated by taking the average inference time per input pair over all test samples. On VIGOR, SliceMatch achieves an FPS of 167, which is considerably faster than global descriptor-based baselines: 50 FPS for CVR [127] for localization only, 29 FPS / 3 FPS for MCC [31] for localization only / pose estimation. On KITTI, SliceMatch runs at 156 FPS, while the local feature-based iterative method, LM [130] has 0.59 FPS. Importantly, the runtime of SliceMatch remains nearly constant as the number of used candidate poses K increases (experiment tested K up to1×10^6).

5.4 Conclusion of the chapter

This chapter has introduced SliceMatch, a novel, accurate, and efficient method for cross-view 3-DoF camera pose estimation. By splitting the HFoV into slices, the proposed architecture can learn discriminative features in terms of both localization and orientation estimation. The proposed aggregation can select the relevant aerial image features for each ground view slice through cross-view attention, and it is observed that there are further accuracy gains by reweighing the terms in the infoNCE loss. With the same VGG backbone, SliceMatch achieves 19% and 62% lower median localization error than the previous state-of-the-art on the VIGOR and KITTI datasets. A better backbone improves SliceMatch's performance even further, e.g. with ResNet50 its 50% lower median error on VIGOR sets a new state-of-the-art. To construct the global descriptor for a candidate pose, only an efficient weighted averaging over the aerial features is needed using precomputed masks (which represent the ground camera's frustum geometry in the aerial view), achieving inference at more than 150 FPS. SliceMatch can include available priors in its candidate poses, e.g. for an initial orientation estimate, but does not require it.

6

Adapting fine-grained cross-view localization to areas without fine ground truth

This chapter is based on *Z. Xia, Y. Shi, H. Li, and J. F. P. Kooij, "Adapting Fine-grained Cross-view Localization to Areas without Fine Ground Truth," European Conference on Computer Vision, 2024 [34].*

6.1 Overview

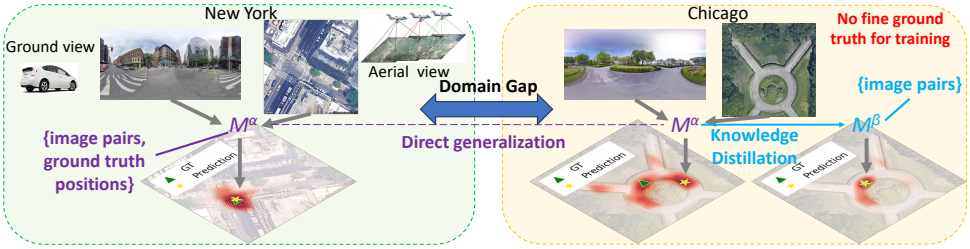


Figure 6.1: Learning-based cross-view localization models often perform well when test images are from the same area used in training, as shown in the green box. When inference in a new target area where no ground truth is available, the standard practice (in purple) directly deploys a model trained in a different area, leaving an obvious domain gap. Due to this domain gap, the direct generalization often results in a performance drop, causing uncertain or erroneous predictions. This chapter proposes a knowledge self-distillation-based weakly-supervised learning approach (in cyan) to adapt the model to the target area using only ground-aerial image pairs without ground truth locations, and this leads to better localization performance.

The key underlying assumption of fine-grained cross-view visual localization [31–33, 130, 131, 134, 163] is that although the accurate fine-grained location of the ground camera is not known, a *coarse* localization prior is available at inference time to identify the aerial image that coves the ground camera's location.

As shown in Figure 6.1, there are two main scenarios in cross-view localization. **(1)** Same-area testing (Figure 6.1, green box): When the fine ground truth, i.e. the accurate location of the ground camera, is available in the target area, a cross-view localization model can be trained on this data and then deployed for inference on new test images. **(2)** Cross-area testing (Figure 6.1, yellow box, left): When there is no fine ground truth in the target area, it is common to train the model on images from a different area for which fine ground truth is available, and then the trained model is directly deployed in the target area. Because of the domain gap between the two areas, the predicted location becomes less reliable. Although many works [31–33, 127, 130, 134, 163] have been proposed for fine-grained cross-view localization, they all suffer from this performance drop when directly deploying in a new target area. Nevertheless, this cross-area scenario is more realistic for real-world use cases, since collecting fine ground truth of every region is expensive and sometimes infeasible. Recent works [33, 130, 134] even found errors in ground truth locations in popular datasets [20, 127, 164, 165]. Therefore, an alternative to fully-supervised training on fine ground truth is needed to scale cross-view localization models to larger areas.

This chapter proposes to address this problem of cross-area localization by relying on the exact same key assumption in the fine-grained cross-view localization task. Namely, it is straightforward to collect ground images with coarse ground truth, i.e. the rough location of the ground camera, at a new area to identify the local aerial image patch. For instance, inaccurate GNSS measurements in urban canyons are unreliable as fine ground truth [140], but can still be used as coarse localization prior. Then, the goal is to *improve a pre-trained model's localization performance in the target area by leveraging only the ground-aerial image pairs in the target area, without associated fine ground truth locations*[1].

[1]Recent models need the ground camera's orientation for training. This chapter assumes the camera orientation

For this goal, this chapter adopts knowledge self-distillation [166, 167] to finetune a fine-grained cross-view localization model in a weakly-supervised manner in which only coarse location is used for pairing the ground and aerial images. A pre-trained model from another area is used as the teacher model to generate pseudo ground truth for the target-area images and use it to train a student model, which is initialized as a copy of the teacher model. Since the teacher's output can be uncertain in the target area, directly using it as pseudo ground truth might reinforce incorrect localization estimates and lead to sub-optimal results. This chapter addresses this by introducing methods to reduce the uncertainty and filter out the outliers in the pseudo ground truth. Concretely, the main contributions of this chapter are:

(1) This chapter proposes a knowledge self-distillation-based weakly-supervised learning approach that considerably improves models' localization performance in a new area by only leveraging the ground-aerial image pairs without ground truth locations. The proposed approach is validated on two state-of-the-art methods on two benchmarks. **(2)** For methods with coarse-to-fine outputs, this chapter investigates how to reduce the uncertainty and suppress the noise in teacher model's predictions. Using the proposed single-modal pseudo ground truth leads to a better student model than using the multi-modal heat maps from the teacher model. **(3)** This chapter designs a simple but effective method for filtering outliers in the pseudo ground truth. Training with filtered pseudo ground truth further improves the localization accuracy of the student model.

6.2 Methodology

This section first formalizes the task of fine-grained cross-view localization. After that, the proposed approach is introduced.

6.2.1 Task definition

Given a ground-level image G and an aerial image A that covers the local surroundings of G, the task of fine-grained cross-view localization is to determine the image coordinates $\hat{y} = (\hat{u}, \hat{v})$ of the ground camera within A, where $\hat{u} \in [0,1]$ and $\hat{v} \in [0,1]$. Recent methods [31–33, 134, 163] achieve this task by training a deep model $\mathcal{M}(G,A)$ which predicts a *heat map* H to capture the underlying localization confidence over spatial locations, and the most confident location can be used as predicted location y,

$$H = \mathcal{M}(G,A), \quad y = \arg\max_{u,v}(H(u,v)). \tag{6.1}$$

To optimize the model's parameters θ_α with respect to a model specific loss functions $\mathcal{L}_\mathcal{M}$, an annotated dataset of a set of N_α ground-aerial image pairs, $\mathbb{I}_\alpha = \{\{G_1, A_1\}, ..., \{G_{N_\alpha}, A_{N_\alpha}\}\}$, and their corresponding ground truth $\mathbb{Y}_\alpha = \{\hat{y}_1, ..., \hat{y}_{N_\alpha}\}$ is used,

$$\theta_\alpha = \arg\min_\theta \mathbb{E}\left[\mathcal{L}_\mathcal{M}(\mathcal{M}(G,A \mid \theta), \hat{y})\right], \tag{6.2}$$

where $\{G,A\} \in \mathbb{I}_\alpha$ and $\hat{y} \in \mathbb{Y}_\alpha$.

is known since it can be acquired easily, e.g. by the digital compass in a mobile phone or a vehicle.

The training image set $\mathbb{I}_\alpha$ consists of samples drawn from a true distribution $\mathcal{D}_\alpha$ representing a specific geographic area α, i.e. $\mathbb{I}_\alpha \overset{\text{i.i.d.}}{\sim} \mathcal{D}_\alpha$. When the model is deployed, the test image set $\mathbb{I}_{test}$ can either come from the *same area* α, or a new environment β. As motivated before, this chapter focuses on the *cross-area* setting, namely $\mathbb{I}_{test}$ is from the target area β, i.e. $\mathbb{I}_{test} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_\beta$. Because of the domain gap, $\mathcal{D}_\beta \neq \mathcal{D}_\alpha$, directly deploying the trained model $\mathcal{M}^\alpha := \mathcal{M}(\cdot \mid \theta^\alpha)$ on $\mathbb{I}_{test}$ as in current practice is sub-optimal.

It is important to note that standard fine-grained cross-view localization [32, 33, 134, 163] assumes the pairing between ground and aerial images is known during inference, as collecting ground-level images with coarse location estimates in the target area is often easy. Therefore, this section proposes to consider the easily available pairing information for weakly-supervised learning by collecting another set of images $\mathbb{I}_\beta = \{\{G_1, A_1\}, ..., \{G_{N_\beta}, A_{N_\beta}\}\}$ from the target area β, $\mathbb{I}_\beta \overset{\text{i.i.d.}}{\sim} \mathcal{D}_\beta$, without corresponding fine ground truth $\mathbb{Y}_\beta$. As noted before, the orientation of the ground camera is assumed known.

The objective is then to adapt a fine-grained cross-view localization model $\mathcal{M}^\alpha$ to the target area β by leveraging the image set $\mathbb{I}_\beta$ without ground truth locations such that the model performance on $\mathbb{I}_{test}$ can be improved.

6.2.2 UDA for cross-view localization

So far, no prior work addressed the task of adapting fine-grained cross-view localization to new areas without labels. To decide on a suitable UDA approach, this section first notes that heat maps of state-of-the-art models reflect more uncertainty for cross-area samples than for same-area samples [32, 33, 163]. The higher uncertainty results in more small positional errors, but also more modes in the heat map, yielding more outliers with large positional errors.

This section therefore considers UDA techniques that can help reduce the uncertainty. One option is *entropy minimization* [168], namely to directly deploy the trained model $\mathcal{M}^\alpha$ on the image set $\mathbb{I}_\beta$ and then encourage the final output heat map H to be more certain by minimizing its entropy. However minimizing the entropy does not necessarily encourage the model to converge towards the correct location for $\{G, A\} \in \mathbb{I}_\beta$, as the model may just as well become more confident about the outliers. Our experiments shall validate entropy minimization's shortcomings for our task.

This section instead proposes to pursue *knowledge self-distillation* [169] for the target task. The trained model $\mathcal{M}^\alpha$ from the source area α can be used as the teacher model to generate *pseudo ground truth* X for image set $\mathbb{I}_\beta$ to train a student model $\mathcal{M}^\beta$. Here, this section considers X as a target heat map with the same spatial resolution as the aerial image A. The student model has the same architecture as the teacher model and is initialized using the teacher model's weights θ_α. Encouraging the outputs of the student model to mimic X can improve the accuracy of the student model on images from β, especially if the generation of pseudo ground truth is controlled to suppress unwanted modes and select for reliable samples.

Finally, this section points out that the recent state-of-the-art methods [32, 163] have K coarse-to-fine heat map outputs, i.e. $\mathbb{H} = \mathcal{M}(G, A)$ and $\mathbb{H} = \{H_1, ..., H_K\}$. The spatial resolution of the next level heat map is higher than that of the previous level, namely $\text{res}(H_{k+1}) > \text{res}(H_k)$ where k is the index for the level and res() returns the spatial resolution.

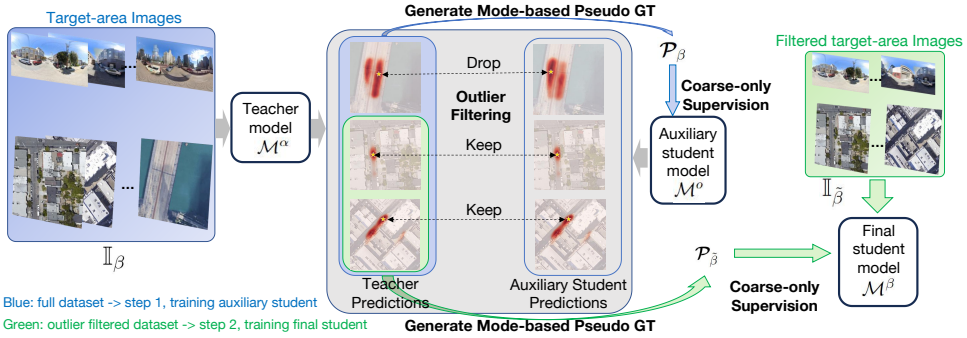


Figure 6.2: Overview of the proposed weakly-supervised learning approach. The proposed method first employs a teacher model trained on data from another area to generate pseudo GT on target-area images, shown in blue. The pseudo GT is then used to train an auxiliary student model $\mathcal{M}_o$. After that, it compares the predictions from the teacher model and those from the auxiliary student model, and filters out unreliable teacher predictions (the middle part of this figure). The remaining predictions, shown in green, are used to train the final student model $\mathcal{M}_\beta$.

The final predicted location then becomes $y = \arg\max_{u,v}(H_K(u,v))$. For other applications with coarse-to-fine models, encouraging shallower layers' activation to mimic deeper layers' activation can bootstrap model performance [169]. Similarly, knowledge self-distillation for cross-view localization may also exploit such coarse-to-fine maps.

6.2.3 Proposed approach

Usually, the deeper layers in the model have access to more information than the shallower layers, e.g. the fine-grained scene layout information passed by the skipped connections, as in UNet [88]. Hence, the output from deeper layers can be more precise than that from shallower layers. This section therefore proposes to follow the "Best Teacher Distillation" paradigm [169] and generate pseudo ground truth X from only the highest-resolution heat map predicted by the teacher model on the target domain input.

A naive approach is using simply $X := H_K^\alpha$ from teacher output[2] $\{H_1^\alpha, \cdots, H_K^\alpha\} = \mathcal{M}^\alpha(G, A)$ for any $\{G, A\} \in \mathbb{I}_\beta$. Then, this high-resolution pseudo ground truth X is down-sampled to create a set of pseudo ground truth heat maps $\mathbb{P} = \{P_1, ..., P_K\}$ to supervise the student model at all levels,

$$P_k = \text{downsample}_k(X) \quad \text{s.t.} \quad \text{res}(P_k) = \text{res}(H_k). \tag{6.3}$$

The set $\boldsymbol{\mathcal{P}}_\beta = \{\mathbb{P}_1, ..., \mathbb{P}_{N_\beta}\}$ is the complete pseudo ground truth for image set $\mathbb{I}_\beta$ in the target area for training the student model, where N_β is the number of the ground-aerial image pairs in $\mathbb{I}_\beta$.

However, since the pseudo ground truth X contains errors, directly following this naive approach might propagate the errors to the student model $\mathcal{M}^\beta$. Thus, this section presents several strategies to reduce the teacher's uncertainty, and deal with noise and large outliers in X. The proposed designs are highlighted in the overview of the approach in Figure 6.2.

[2]Note that this chapter uses superscript α to indicate output from model $\mathcal{M}^\alpha$.

Coarse-only Supervision: Standard Best Teacher Distillation [169] suggests supervising heat maps at all levels of the student model using the pseudo ground truth. However, the spatial accuracy of X is limited, and using X to supervise the high-resolution outputs of the student model might propagate this noise. It is noted that the down-sampling in Equation 6.3 suppresses such positional noise at the lower resolution P_k. Thus using only the lower level P_k might lead to a better student model. This section therefore considers to only compute the loss on student model's outputs $\mathbb{H}^\beta = \mathcal{M}^\beta(G, A)$ up to a certain level $K' \leq K$,

$$\mathcal{L}(\mathbb{H}^\beta, \mathbb{P}) = \frac{1}{K'} \sum_{k=1}^{K'} \mathcal{L}_k(H_k^\beta, P_k). \tag{6.4}$$

Here K' is a hyperparameter, and $\mathcal{L}_k(H_k^\beta, P_k)$ is a weighted sum of infoNCE losses [154], similar to regular training in [31, 32], except this section uses pseudo ground truth P_k as weight,

$$\mathcal{L}_k(H_k^\beta, P_k) = \frac{1}{\sum P_k} \sum_{m,n} P_k^{m,n} \cdot \mathcal{L}_{\text{infoNCE}}(H_k^\beta \mid (m,n)). \tag{6.5}$$

$\mathcal{L}_{\text{infoNCE}}(H_k^\beta \mid (m,n))$ denotes an infoNCE loss interpreting H_k^β as metric learning scores, location (m,n) as the positive class, and all other locations as the negative class.

Mode-based Pseudo Ground Truth: Rather than using H_K^α directly as pseudo ground truth X, This section proposes to create a "clean" pseudo ground truth X that only represents its mode $y^\alpha = \arg\max(H_K^\alpha)$. This section thus provides the student with a training objective that represents less uncertainty for the target domain input than its teacher. Still, it is common when training fine-grained cross-view localization models, to apply Gaussian label smoothing [31, 134] even with reliable ground truth to aid the learning objective and increase robustness to remaining errors in the annotation [170]. This section similarly applies Gaussian label smoothing centered at y^α,

$$X(u,v) = \mathcal{N}((u,v) \mid y^\alpha, I_2\sigma^2), \ \text{res}(X) = \text{res}(A). \tag{6.6}$$

In Equation 6.6, the standard deviation σ is a hyperparameter and I_2 is a 2D identity matrix.

Outlier Filtering: Recent deep learning advances [171] highlighted the importance of using curated data. Motivated by this principle, this section prefers having fewer but more reliable samples of the target domain, over having more samples but with potentially large errors in the pseudo ground truth. The *Mode-based Pseudo Ground Truth* could force a sample's ground truth to commit to a wrong (outlier) location, therefore this section seeks to filter out such samples.

This section here makes another observation: samples where the predicted locations y^α of a teacher and y^β of a student greatly differ, the teacher's predictions were more likely to be outliers compared to samples where the teacher and student's predicted locations are more consistent, as it will be demonstrated in the experiments. Thus, this section proposes to first train another auxiliary student model $\mathcal{M}^o$ on all data from the target domain, and compare its prediction to the teacher's to identify stable predictions with little change in

the predicted location. Then, only those reliable non-outlier samples are used to train the final student model $\mathcal{M}^{\beta}$.

Concretely, this section first optimizes the auxiliary student model $\mathcal{M}^{o}$ on all $\mathbb{I}_{\beta}$ with $\mathcal{P}_{\beta}$ using,

$$\theta_o = \arg\min_{\theta} \mathbb{E}[\mathcal{L}(\mathcal{M}(G, A \mid \theta), \mathbb{P})], \tag{6.7}$$

where $\{G, A\} \in \mathbb{I}_{\beta}$ and $\mathbb{I}_{\beta} \in \mathcal{P}_{\beta}$.

Then, this section calculates the L2-distance $d^{\alpha,o} = \|\mathbf{y}^{\alpha} - \mathbf{y}^{\mathbf{o}}\|_2$ between the image coordinates predicted by $\mathcal{M}^{\alpha}$ and $\mathcal{M}^{o}$ to find the potential unreliable $\mathbb{P}$. The resulting distance set $\mathbb{D} = \{d_1^{\alpha,o}, ..., d_{N_{\beta}}^{\alpha,o}\}$ is used to keep the top-T% samples in $\mathbb{I}_{\beta}$ that have the smallest T% distance $d^{\alpha,o}$. Denoting the filtered image set as $\mathbb{I}_{\tilde{\beta}}$ and corresponding pseudo ground truth as $\mathcal{P}_{\tilde{\beta}}$, the final student model $\mathcal{M}^{\beta}$ is optimized using Equation 6.7 by substituting $\mathbb{I}_{\beta}$ with $\mathbb{I}_{\tilde{\beta}}$ and $\mathcal{P}_{\beta}$ with $\mathcal{P}_{\tilde{\beta}}$.

6.3 Experiments

This section first introduces the two used datasets and the evaluation metrics. Then, it discusses two state-of-the-art methods [32, 163], based on which the proposed weakly-supervised learning is evaluated, followed by implementation details. After this, the test results and a detailed ablation study are provided.

6.3.1 Datasets

This chapter adopts two cross-view localization datasets, VIGOR [127] and KITTI [20], and focuses on their cross-area split.

VIGOR dataset contains ground-level panoramic images and their corresponding aerial images collected in four US cities. In its cross-area split, the training set contains images from two cities, and the test set is collected from the other two cities. This section uses the training set to train the teacher model and focuses on the cross-area setting in the experiments. To compare direct generalization and our proposed weakly-supervised learning, we conduct a 70%, 10%, and 20% split on the original cross-area test set to create our weakly-supervised training set (no ground truth locations), validation set, and test set. This section uses the validation set for finding the stopping epoch during training, as well as for conducting the ablation study. The test set is used for benchmarking the proposed method. The improved VIGOR labels provided by [33] are used.

KITTI dataset contains ground-level images with a limited field of view. Experiments use the aerial images provided by [130] and adopt their cross-area setting, where the training and test images are from different areas. Similar to the settings on the VIGOR dataset, this section uses the training set to train the teacher model and then splits the original cross-area test set into 70%, 10%, and 20% for weakly-supervised training of the student model, validation, and testing.

6.3.2 Evaluation metrics

THis section measures the displacement error ϵ in meters between the predicted location and the ground truth location, i.e., $\epsilon = s\|y - \hat{y}\|_2$, where s is the scaling factor from image

coordinates to real-world Euclidean coordinates. Then, mean and median displacement errors over all samples are reported as our evaluation metrics. Since ground-level images in the KITTI dataset have a limited field of view, this section further decomposes their displacement errors into errors in the longitudinal direction (along the camera's viewing direction, typically along the road), and errors in the lateral direction (perpendicular to the viewing direction).

6.3.3 Baseline state-of-the-art methods

Two state-of-the-art methods, Convolutional Cross-View Pose Estimation (CCVPE) [32] and Geometry-Guided Cross-View Transformer (GGCVT) [163] are used to test our proposed weakly-supervised learning approach. Both methods were proposed for fine-grained cross-view localization and orientation estimation, and have a coarse-to-fine architecture. CCVPE has two separate branches for localization and orientation prediction. GGCVT uses an orientation estimation block before its location estimator. This section uses them for localization only. CCVPE has seven levels of heat map outputs, in which the first six heat maps are 3D, with the first two dimensions for localization and the third dimension for orientation. The last heat map is 2D. GGCVT has three levels of 2D heat map outputs.

6.3.4 Implementation details

This section uses the code released by the authors of CCVPE [32] and GGCVT [163] for model implementations. Student and auxiliary models are trained following our proposed approach. For CCVPE's 3D heat map output, this section simply lifts the pseudo ground truth heat map P_k to 3D using the known orientation as done in [32]. Following the two model's default settings, this section uses a batch size of 8 for CCVPE and 4 for GGCVT, and a learning rate of 1×10^{-4} with Adam optimizer [147] for both models.

The hyperparameters K', T, and σ are tuned on the VIGOR validation set. For CCVPE, it is found that including the first two levels of losses, i.e. $K' = 2$, and $T\% = 80\%$ gives the lowest mean localization error. For GGCVT, all three levels of losses are used, i.e. $K' = 3$, and $T\% = 70\%$. This section uses $\sigma = 4$ (pixels) for both methods. The same setting is directly applied to KITTI.

6.3.5 Results

This section compares the trained student models to teacher models (baselines) on the cross-area test set of VIGOR and KITTI datasets. Previous state-of-the-art was set by directly deploying CCVPE and GGCVT teacher models to the target area. On the VIGOR dataset, Table 6.1 top, the performance of student models trained using proposed weakly-supervised learning surpasses baselines by a large margin. For CCVPE, the proposed approach reduces the mean and median error by 20% and 15% when the orientation of test ground images is unknown. GGCVT only released its code and models for orientation-aligned ground-aerial image pairs for the VIGOR dataset. Thus, this section follows the same setting. In this case, the proposed approach reduces 16% and 5% mean and median error for GGCVT. Without extra hyperparameter tuning, this section directly uses the proposed approach to train models on the KITTI dataset, and it again improves the overall localization performance for both models, see Table 6.1 bottom.

VIGOR, cross-area test	Known orientation		Unknown orientation	
	Mean (m)	Median (m)	Mean (m)	Median (m)
CCVPE [32]	4.38	1.76	5.35	1.97
CCVPE student (proposed)	**3.85** (↓ 12%)	**1.57** (↓ 11%)	**4.27** (↓ 20%)	**1.67** (↓ 15%)
GGCVT [163]	5.19	1.39	-	-
GGCVT student (proposed)	**4.34** (↓ 16%)	**1.32** (↓ 5%)	-	-

KITTI, cross-area test	Longitudinal error		Lateral error	
	Mean (m)	Median (m)	Mean (m)	Median (m)
CCVPE [32]	6.55	2.55	1.82	**0.98**
CCVPE student (proposed)	**6.18** (↓ 6%)	**2.35** (↓ 8%)	**1.76** (↓ 3%)	**0.98** (↓ 0%)
GGCVT [163]	9.27	4.66	2.19	0.85
GGCVT student (proposed)	**8.56** (↓ 8%)	**4.35** (↓ 7%)	**1.90** (↓ 13%)	**0.79** (↓ 7%)

Table 6.1: Evaluation on VIGOR and KITTI test set. **Best in bold.** Baseline models are teacher models (previous state-of-the-art). "Student" denotes models trained using our proposed weakly-supervised learning without ground truth labels. On VIGOR, test results for both known and unknown orientation cases are provided. On KITTI, models are tested with known orientation.

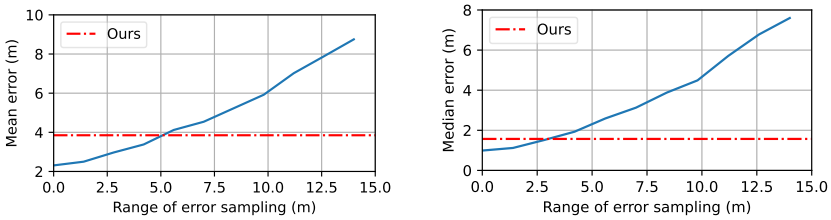


Figure 6.3: VIGOR test set errors (vertical axis) of CCVPE models fintuned on noisy ground truth. The horizontal axis denotes the upper bound for error sampling. Ours: the student model trained with the proposed weakly-supervised learning.

This section also studies the gap between each student model to an Oracle, i.e. the same method using supervised finetuning on fine ground truth at the target area. Even though the Oracles still achieve lower errors (CCVPE: Oracle 2.31 m *vs.* student 3.85 m; GGCVT: Oracle 2.91 m *vs.* student 4.34 m), it is emphasized that, in practice, such reliable fine ground truth is generally not available. Importantly, this section also finds that when the ground truth does contain errors, using supervised finetuning leads to large test errors, see Figure 6.3. Instead, the proposed weakly-supervised learning approach scales well because it boosts performance at a low cost: First, there are no extra requirements on the accuracy of localization prior in the target area over previous fine-grained cross-view localization works [31–33, 131, 134, 163], as only ground-aerial image pairs are needed. Second, since student models are initialized from their teacher, the training time is short. For example, on VIGOR, using a single 32GB V100 GPU our weakly-supervised learning for CCVPE only adds ~ 6 hours of training time (including pseudo ground truth generation and outlier filtering) on top of the direct generalization, which has training time of ~ 16 hours.

Next, two samples where the student model improves over the teacher model are visualized. A typical case is shown in Figure 6.4 left, in which the teacher model has a multi-modal prediction, and the peak is located in a wrong mode. The student model learned to weigh the modes better after adapting to the target environment. As shown

Figure 6.4: CCVPE teacher and student model's predictions on VIGOR test set. The red color denotes the localization probability (a darker color means a higher probability).

in the example on the right in Figure 6.4, sometimes, even though the teacher model's heat map does not capture the correct location, the student model can still identify it. In this case, the student model might learned discriminative features from other samples in this area to localize the ground camera. This demonstrates the effectiveness of the knowledge-distillation process for cross-area inference (more examples are included in Figure 6.5).

6.3.6 Analysis of prediction errors after KD

Following the visual examples, the overall statistical relation between the model prediction errors, and the change in predicted locations after knowledge distillation are now analyzed. Figure 6.6 and 6.7 plot this relation for CCVPE.

First, it is confirmed that potential outliers can indeed be identified by the amount of difference between the predicted locations of a teacher and its auxiliary student model in Figure 6.6 left. It can be seen that there is a large portion of samples located around the

Figure 6.5: Teacher and student models' predictions on VIGOR test set. The red color denotes the localization probability (a darker color means a higher probability). First three: success cases. Last: a failure case.

diagonal line, i.e. $\epsilon^{\alpha} = s \cdot d^{\alpha,o}$. Most samples in $\mathbb{I}_{\alpha}$ with large change $d^{\alpha,o}$ in predicted location indeed obtained a large error ϵ^{α} for the teacher model's prediction. Next, Figure 6.6 right shows how the difference in location correlates with the prediction error of the auxiliary student. There are more samples being scattered at the bottom of the plot, implying many wrong predictions of the teacher model have already been corrected. Still, our ablation study will demonstrate that using the auxiliary student model directly as a new teacher for a final student model does not work as well as using it for outlier detection. Note that the (less prominent) diagonal line now indicates errors introduced by the auxiliary student model. Then, this section validates that the final student model reduces the localization error compared to the teacher model on the target test set $\mathbb{I}_{test}$ in Figure 6.7. Comparing the left plot to the right plot, this section observes a similar trend as for the auxiliary student model before, namely that the many samples with high teacher error in the left plot now obtain low student error in the right plot. The same trend can be observed for GGCVT models in Figure 6.8 and 6.9.

Lastly, this section compares the error in predictions of the teacher model and that of the student model for both CCVPE and GGCVT on the VIGOR test set $\mathbb{I}_{test}$. The error change after weakly-supervised knowledge self-distillation is calculated and its statistics are visualized in Figure 6.10. The left part of the two histograms (in purple and magenta) shows the samples that have a smaller error in the student model's prediction. Similarly, the right part of the two histograms (in navy and orange) denotes the samples that the teacher model has a more accurate prediction. Overall, it can be seen that, for both CCVPE and GGCVT, there are more samples located in the left part. It demonstrates that the student model reduces the error for the majority of samples.

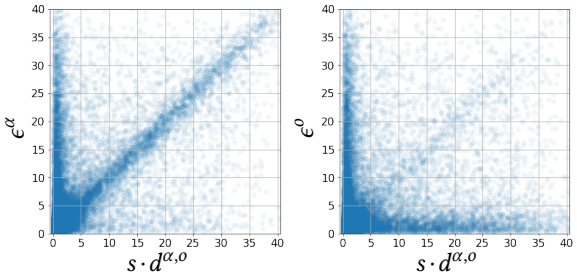


Figure 6.6: Teacher (left) v.s. Auxiliary student (right) models on $\mathbb{I}_{\beta}$. CCVPE model, relation between error ϵ and change d in predicted locations from teacher and auxiliary student models on VIGOR. ϵ^{α} / ϵ^{o}: errors (m) of teacher model's / auxiliary student model's predictions. $s \cdot d^{\alpha,o}$: the difference (m) between predicted locations of teacher and auxiliary student.

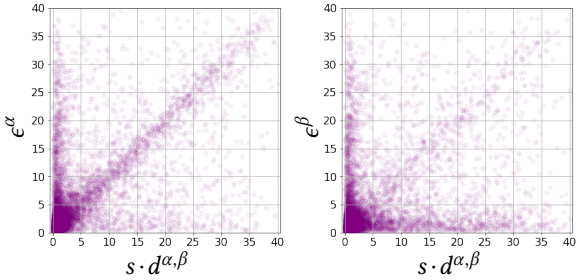


Figure 6.7: Teacher (left) v.s. Final student (right) models on $\mathbb{I}_{test}$. CCVPE model, relation between error ϵ and change d in predicted locations from teacher and final student models on VIGOR. ϵ^{α} / ϵ^{β}: errors (m) of teacher model's / final student model's predictions. $s \cdot d^{\alpha,\beta}$: the difference (m) between predicted locations of teacher and final student.

6.3.7 Domain adaptation by entropy minimization

This section tests entropy minimization [168] for the CCVPE model on the VIGOR dataset as an alternative technique to adapt a model from the source domain to the target domain. Entropy minimization is often used for semi-supervised domain adaptation [172]. In this setting, the model is trained with a combination of samples with ground truth labels from the source domain and unlabeled samples from the target domain. When a source domain sample is presented, the model is trained using its default supervised learning loss $\mathcal{L}_{\mathcal{M}}$. When the input is from the target domain, the training objective is to minimize the entropy of the output prediction using an entropy minimization loss $\mathcal{L}_{EM}$.

This experiement trains a CCVPE model [32] on the VIGOR dataset using loss $\mathcal{L}_{final}$,

$$\mathcal{L}_{final} = \begin{cases} \mathcal{L}_{\mathcal{M}}(\mathcal{M}(G,A),\hat{y}), & \text{if } \{G,A\} \in \mathbb{I}_{\alpha}, \hat{y} \in \mathbb{Y}_{\alpha}, \\ \omega \cdot \mathcal{L}_{EM}(H_K), & \text{if } \{G,A\} \in \mathbb{I}_{\beta}. \end{cases} \tag{6.8}$$

In Equation 6.8, $\mathcal{L}_{\mathcal{M}}$ is the default loss in [32], H_K is the final output heat map of the model $\mathcal{M}$ on image pair $\{G,A\}$, and ω is a hyperparameter that weighs the entropy minimization loss $\mathcal{L}_{EM}$. As in [172], the pixel-wise Shannon Entropy [173] in the dense output is

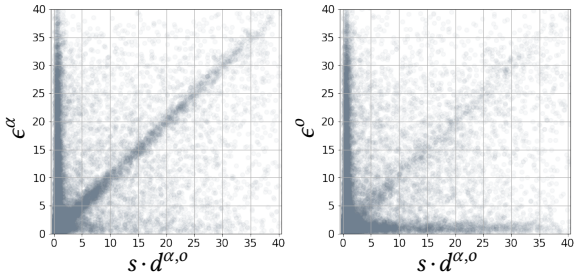


Figure 6.8: Teacher (left) v.s. Auxiliary student (right) models on $\mathbb{I}_{\beta}$. GGCVT model, relation between error ϵ and change d in predicted locations from teacher and auxiliary student models on VIGOR. ϵ^{α} / ϵ^{o}: errors (m) of teacher model's / auxiliary student model's predictions. $s \cdot d^{\alpha,o}$: the difference (m) between predicted locations of teacher and auxiliary student.

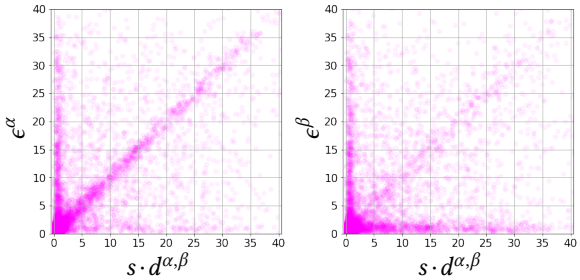


Figure 6.9: Teacher (left) v.s. Final student (right) models on $\mathbb{I}_{test}$. GGCVT model, relation between error ϵ and change d in predicted locations from teacher and final student models on VIGOR. ϵ^{α} / ϵ^{β}: errors (m) of teacher model's / final student model's predictions. $s \cdot d^{\alpha,\beta}$: the difference (m) between predicted locations of teacher and final student.

calculated, and then the sum of all pixel-wise entropy is used as the $\mathcal{L}_{EM}$,

$$\mathcal{L}_{EM}(H_K) = -\sum_{u,v} H_K(u,v) \cdot \log(H_K(u,v)), \tag{6.9}$$

$H_K(u,v)$ denotes the value at each location in the output heat map H_K.

This section tuned ω and found that joint training with entropy minimization always hurts the model performance. As shown in Figure 6.11, the mean and median error on the validation set (target area) increases when the model is trained using a larger weight ω, and the best model appears when $\omega = 0$, equivalent to direct generalization of a model trained in a supervised manner on only source domain images.

For completeness, this section also tried directly fine-tuning a pre-trained model from the source domain on images from the target domain using entropy minimization (no joint supervised training with source domain samples). Since the model failed completely, the plots are not included.

Entropy minimization simply encourages the heat map to be sharper in the target area. Therefore, it does not resolve multi-modal uncertainty. As shown in Figure 6.12, compared to direct generalization, training with entropy minimization makes the red region in the

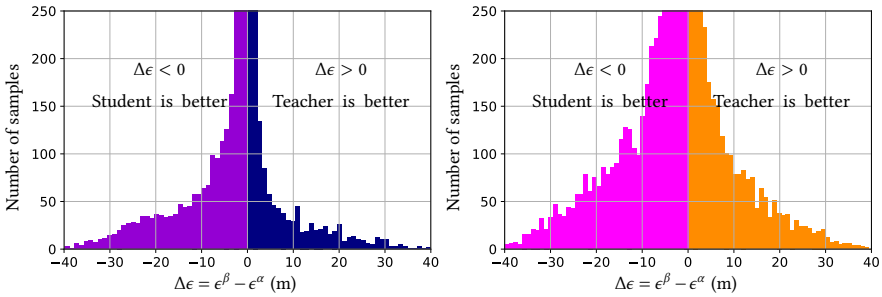


Figure 6.10: Change in error between predictions of the teacher $\mathcal{M}^{\alpha}$ and those of the student model $\mathcal{M}^{\beta}$ on VIGOR test set $\mathbb{I}_{test}$. Purple and Magenta region: The student model has smaller errors. Navy and Orange region: The teacher has smaller errors. Left: CCVPE model, right: GGCVT model.

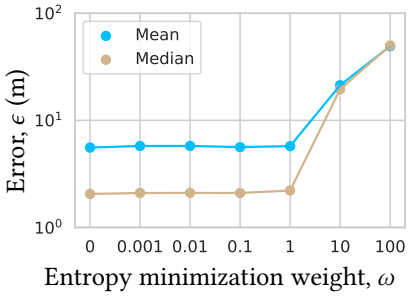


Figure 6.11: Errors of CCVPE models with different entropy minimization weights ω on VIGOR validation set.

heat map smaller, but the peak of the heat map stays in the same mode in the multi-modal distribution. Instead, our proposed knowledge self-distillation adapts the model to the target domain by explicitly encouraging the model to disambiguate multiple modes using the proposed single-modal pseudo ground truth. As a result, our proposed method can correct the wrong mode and also reduce uncertainty.

Therefore, simply exposing the model to the images from the target area and enforcing the confidence of outputs is not sufficient for improving cross-view localization across areas. The proposed knowledge self-distillation instead reduces uncertainty by filtering out unreliable samples.

6.3.8 Domain adaptation by other pseudo label approaches

The proposed Coarse-only Supervision uses the model's high-resolution output to supervise low-resolution ones. Alternatively, this section also studies fusing the outputs at different levels to generate supervision signals.

Similar to [174], this section fuses information in both top-down and bottom-up directions to generate pseudo ground truth at each level for the student model. It is achieved by up/downsampling teacher's matching volumes at different levels and fusing them with averaging. The error of the resulting student (4.49 m) is larger than ours (3.85 m). A hy-

Figure 6.12: Adapting a CCVPE model to the target domain with different methods. Results on the VIGOR test set. Comparison between direct generalization (No EM, $\omega = 0$), different entropy minimization weights (EM, $\omega = 0.1$ and EM, $\omega = 1.0$), and the proposed knowledge self-distillation (KD, ours). The red color denotes the localization probability (a darker color means a higher probability).

pothesize is that, for localization, fine-grained high-resolution heatmaps can help supervise low-resolution maps, but not vice versa, which may be why [174]'s top-down + bottom-up approach does not work well for our task.

As an alternative to the proposed outlier filtering, this section also tries an uncertainty-based outlier filtering approach while keeping other proposed modules unchanged. Similar to [175–177], the entropy of teacher's output heat maps is used as a measure of their uncertainty. The teacher's heatmaps are ranked based on their entropy and then the most certain $T\%$ is used for student training. For a fair comparison, CCVPE uses top 80% and GGCVT uses top 70% (same as in the proposed outlier detection). The resulting models have higher errors (CCVPE/GGCVT: 4.17/4.52 m) than the proposed ones (3.85/4.34 m). Entropy-based methods do not consider the spatial order of classes, e.g. a two-mode heatmap with 1m between two modes will have the same entropy as a two-mode heatmap with 10m between modes. However, the latter results in larger errors.

6.3.9 Ablation Study

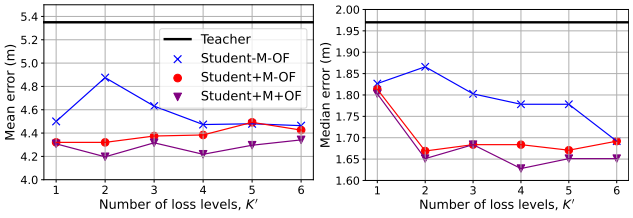


Figure 6.13: Ablation study on the proposed mode-based pseudo ground truth, outlier filtering, and different levels for coarse-only supervision in our teacher-student KD using CCVPE.

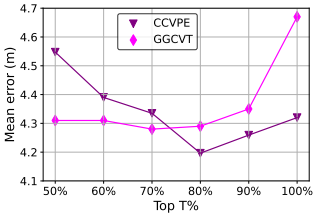


Figure 6.14: Effect of T in the proposed outlier filtering. 100% means no outlier filtering.

An extensive ablation study is conducted to validate the effectiveness of the proposed

designs. The following is denoted: **Teacher** (baseline): directly deploy the teacher model $\mathcal{M}^{\alpha}$ in the target area. **St-M-OF**: student model trained using teacher's heat maps, no mode-based pseudo ground truth, no outlier filtering. **St+M-OF**: student model trained using mode-based pseudo ground truth, no outlier filtering. **St+M+OF** (proposed): student model trained using mode-based pseudo ground truth with outlier filtering, i.e. $\mathcal{M}^{\beta}$.

The performance of these ablation variants when supervising different levels of student predictions of the CCVPE is shown in Figure 6.13. It can be seen that the proposed mode-based pseudo ground truth (+M) and outlier filtering (+OF) both improve the performance, and the final version, St+M+OF, achieves the best results, no matter how many prediction levels of the student model is supervised. For CCVPE student models, supervising the first $K' = 2$ and $K' = 4$ levels have similar localization performance overall. Since $K' = 2$ gives the lowest mean error, it is used in the final setting. Experiments also tuned K' for GGCVT and found that supervising all three levels, i.e. $K' = 3$ gives the best results. The effectiveness of the proposed mode-based pseudo ground truth (+M) and outlier filtering (+OF) on GGCVT is verified in Table 6.2. Additionally, it is also tried to directly use the predictions of the auxiliary student as pseudo ground truth to train the final student model (similar to iterative knowledge self-distillation [166]), denoted as St+M+A in Table 6.2. However, it does not perform better than using the auxiliary student model for outlier filtering.

Error (m)	Teacher	St-M-OF	St+M-OF	St+M+A	St+M+OF
Mean	5.16	5.34	4.67	4.54	**4.28**
Median	1.40	1.48	1.32	1.55	**1.28**

Table 6.2: Ablation study for GGCVT. **Best in bold.**

Figure 6.14 shows the ablation study results on different percentage values T in the proposed outlier detection. The best CCVPE and GGCVT student models appear at $T = 80\%$ and $T = 70\%$. In general, there is a trade-off between the quality and quantity of data. When too little data is kept, there is a risk of model overfitting. Filtering out some detected outliers ($20\% \sim 30\%$) improves the quality of the data and can result in better model performance. This suggests that, in practice, blindly increasing the data amount without guaranteeing its quality might negatively influence models' performance.

6.3.10 t-SNE Feature

To study if the extracted features by the teacher and final student models differ, this section uses t-SNE [178] to map the features to a two-dimensional space for visualization. The CCVPE's ground features and the aerial features at the GT locations at the bottleneck are used. Figure 6.15 shows their t-SNE plots before (teacher model) and after adaptation (final student model). For the teacher model, ground and aerial samples are disjoint in the feature space, complicating matching across views. For the final student model, the plot shows more overlap between the two views, indicating better alignment. This result supports that the quantitative improvement of the proposed approach results from adaptation to the target domain.

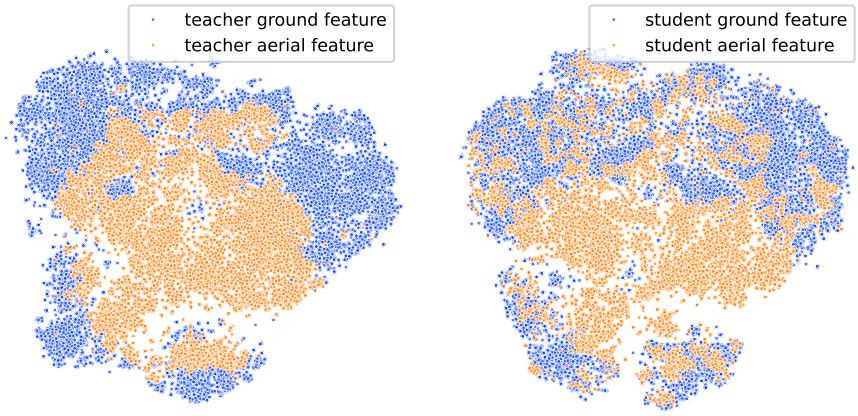


Figure 6.15: t-SNE, VIGOR test set: CCVPE teacher model (left) and final student model(right).

6.3.11 Limitations

In knowledge self-distillation, it is often required that the initial model is at a "good enough" starting point, otherwise, it will not converge to a better solution. This requirement also applies to the proposed method. This section conducts experiments where a teacher model, trained on one dataset such as KITTI [20], was used to generate pseudo ground truth to train a student model on a different dataset, for instance, the Ford dataset [164]. In this case, the teacher's predictions on the target dataset were not much better than random guesses, making the proposed method not applicable. When the training and test sets are from different datasets, the teacher fails in the target area since the domain gap comes not only from different areas, but also from different sensors, and different resolutions of aerial images. This chapter targets the domain gap between different areas but for the same sensor setup.

6.4 Conclusion of the chapter

This chapter focuses on improving the localization performance of a pre-trained fine-grained cross-view localization model in a new target area without accurate ground truth locations. This chapter has proposed a knowledge self-distillation-based weakly-supervised learning approach that only requires a set of ground-aerial image pairs from the target area. Extensive experiments were conducted to study how to generate appropriate pseudo ground truth for student model training. It is found that selecting the predominant mode in the teacher model's predictions is better than directly using the output heat maps. Furthermore, supervising coarse-level predictions of a student model using the down-sampled teacher model's high-resolution predictions can suppress the positional noise and might lead to a slight boost in the student model's performance. Last but not least, this chapter demonstrates that unreliable target domain samples can be filtered out by comparing predicted locations from teacher and student models, which motivates using an auxiliary student model to curate the data. Training a final student model on the

filtered data further improves the localization accuracy. The proposed approach has been validated on two state-of-the-art methods on two benchmarks. It achieves a consistent and considerable performance boost over the previous standard that directly deploys the trained model in the new target area.

7

Conclusion

The preceding chapters of this dissertation have thoroughly explored ground-to-aerial cross-view visual localization, examining various aspects such as problem formulation, methodology, efficiency, and the utilization of training data. This chapter first discusses the key findings from the previous chapters and then answers the research questions. Following that, it presents the core insights derived from the research conducted in this dissertation and concludes with suggestions for future research directions.

7.1 Key findings

In order to determine the most suitable approach for ground-to-aerial cross-view visual localization, Chapters 3 and 4 explored different methodologies. Chapter 3 adopted the common formulation found in previous studies [105, 106, 121], addressing cross-view localization through image retrieval. However, this chapter distinguished itself from earlier works by proposing the integration of localization priors, such as GNSS positioning, into the training of cross-view image retrieval models, rather than training a globally discriminative model. An aerial image patch of a geographical local area, identified by the noisy GNSS localization prior, is densely sampled into smaller, overlapping patches. A model is then trained to retrieve the most similar aerial patch among these geo-local aerial patches for a given ground-level query image. This was achieved by the proposed geo-local triplet loss, which incorporates the geographic distance between samples in a triplet to weigh the corresponding loss. Experiments conducted with two baseline models across two datasets demonstrated that training with the proposed loss significantly enhances model performance in the targeted local area. The proposed loss also encouraged the model to focus on features from objects that are more locally discriminative, a detail overlooked by baselines trained with the standard triplet loss.

However, formulating cross-view localization as an image retrieval problem introduces a trade-off between localization accuracy and computational demand, as achieving high localization accuracy requires densely sampling the aerial image. Chapter 4 proposed a different approach by directly matching the ground-level image with a known aerial image patch, for example, of the size of 70 m^2, that covers the query ground-level image. The CCVPE method, proposed in Chapter 4, significantly reduced the median localization error on the Oxford RobotCar dataset to approximately 1 m. This performance markedly surpasses that of the image retrieval-based method, which, even when combined with temporal filtering and GNSS positioning, resulted in a median localization error of 2.36 m.

To address the challenge of jointly estimating the location and orientation of the ground camera with cross-view image matching, both CCVPE, introduced in Chapter 4, and SliceMatch, presented in Chapter 5, utilized the projection geometry between ground and aerial views. CCVPE uses a fully convolutional ground encoder, leveraging the translational equivariance property of CNNs to preserve heading information from the ground-level image within its encoded 1D ground descriptor. Consequently, each block of elements in the resulting ground descriptor correlates with specific columns of pixels in the input image. The orientation-aware ground descriptor then encourages the aerial descriptor at the ground truth location to also encode orientation information through contrastive learning, meaning that the elements in the aerial descriptor are trained to match those in the ground descriptor, thus becoming orientation-aware. This approach obviates the need for the model to independently learn how to encode orientation information from the data

alone. An ablation study highlighted the benefits of this design choice. Experimental results demonstrated that the CCVPE method significantly surpasses previous state-of-the-art baselines. Specifically, on the VIGOR cross-area test set, CCVPE achieved a median error of 1.89 m and a median orientation error of 13.58° when directly generalized to test data collected in new cities.

Chapter 5 proposed the SliceMatch method, which constructs ground descriptors in a manner similar to that of CCVPE. However, instead of relying on contrastive learning to train the aerial descriptor to be orientation-aware, SliceMatch explicitly multiplies its pre-constructed slice masks with the aerial features to select features for each viewing direction and build aerial slice descriptors. The complete aerial descriptor is then an aggregation of all aerial slice descriptors. SliceMatch's pose estimation is achieved by comparing the ground descriptor with the aerial descriptor constructed at various poses. Although SliceMatch has slightly lower localization accuracy than CCVPE, with a median error of 3.31 m on the VIGOR cross-area test set, it operates significantly faster than CCVPE, achieving 167 FPS compared to 24 FPS. SliceMatch's fast runtime is achieved by leveraging pre-constructed slice masks, allowing operations such as feature selection and aggregation, as well as pose estimation, to be performed through matrix multiplication. This approach trades off runtime efficiency against storage requirements, as it necessitates the storage of extra slice masks. In practice, as outlined in Section 1.2 on localization latency requirements, the runtime is of critical importance. Delays in localization estimation can result in significant location discrepancies, especially when the vehicle travels at high speeds.

Ground-to-aerial cross-view localization models can generalize to new regions, but depending on the domain gap between the training and test regions, there is always a small or considerable performance drop. Chapter 6 focused on improving the scalability of cross-view localization methods to new test areas. It highlighted the challenge of acquiring accurate ground truth locations due to the associated costs, while noting that collecting ground-level images with coarse locations, for example, containing errors of up to tens of meters, is relatively easy using devices like a mobile phone and its built-in GNSS. Consequently, Chapter 6 proposed leveraging noisy ground truth data in the target area to fine-tune a cross-view localization model initially trained in another area with precise ground truth. This approach contrasts with the common practice of directly deploying a model trained in one area to another without fine-tuning. Chapter 6 introduced a knowledge self-distillation-based weakly-supervised learning framework that relies solely on paired ground and aerial images to fine-tune a pre-trained model. It utilized the pre-trained model as a teacher model to generate pseudo ground truth for fine-tuning a student model, which is initialized as a copy of the teacher model. To address small positional errors and filter out significant outliers in the teacher's outputs, Chapter 6 proposed generating single-modal supervision signals to guide the coarse-level outputs of the student model, along with an outlier filtering technique that compares the outputs of the teacher and an auxiliary student model. Experimental results, utilizing two baseline methods across two datasets, demonstrated that fine-tuning the trained model with the proposed framework results in a reduction of up to 16% in both mean and median localization errors.

7.2 Answers to research questions

Based on the key findings derived from Chapter 3 to 6, the research questions can be answered.

7.2.1 Answers to sub-questions:

SQ1: Is the common image retrieval formulation in ground-to-aerial cross-view image matching well-suited for vehicle localization?

The common formulation for image retrieval does not incorporate any localization priors during training. However, consumer-grade vehicles equipped with GNSS receivers can provide a rough localization estimate. When this prior is incorporated during training, the accuracy of image retrieval-based cross-view localization improves. Still, localization for autonomous driving requires methods to be both accurate and efficient (see Section 1.2), and the accuracy and efficiency trade-off brought by image retrieval formulation makes it sub-optimal. As demonstrated in Chapter 4 and 5, directly localizing a ground-level image inside its corresponding local aerial image, identified using the GNSS prior, results in superior localization accuracy with fast runtime. Furthermore, approaching this task as a classification problem, thereby allowing for the modeling of multi-modal uncertainty, produces a more accurate estimate than the single-modal regression approach used in the baseline methodology [127].

SQ2: How can the location and orientation of the ground-level camera be jointly estimated?

The joint estimation of location and orientation can be achieved by leveraging the geometric relationship between ground and aerial views and constructing orientation-aware ground and aerial image descriptors. By matching an orientation-aware ground descriptor to the orientation-aware aerial descriptor at each spatial location, one can obtain a joint distribution for both localization and orientation estimation. The CCVPE and SliceMatch methods, proposed in Chapters 4 and 5 respectively, were developed based on this principle. The difference lies in the construction of the orientation-aware aerial descriptor: CCVPE leverages the power of contrastive learning, while SliceMatch uses pre-constructed slice masks to control feature aggregation during the descriptor-building process.

SQ3: What strategies can be employed to create an efficient ground-to-aerial cross-view visual localization method?

To enhance efficiency, one can consider incorporating the projection geometry between ground-level and aerial images into the deep learning models. Instead of using a large model to learn everything from data, a smaller model can be designed to learn only the information that is not already known. The SliceMatch method, proposed in Chapter 5, serves as an example. It utilizes the geometric relation between ground and aerial views and employs pre-constructed slice masks to simplify the learning process for orientation-aware aerial descriptors. Therefore, no specialized layers are needed for feature aggregation after the backbone feature extractor. The construction of image descriptors is simplified by multiplying the extracted features with pre-constructed slice masks and then averaging the features, a process that can be computed efficiently. As a result, SliceMatch achieved fast runtime, i.e. 156 FPS on the KITTI dataset.

SQ4: Do ground-to-aerial cross-view visual localization methods generalize to new regions, and how can their scalability be enhanced with easily collectable data?

Experiments in Chapter 3 to 5 showed that ground-to-aerial cross-view visual localization methods can generalize to new regions, but similar to other deep learning methods, there is always a performance drop because of the domain gap between training and test regions. The scalability of ground-to-aerial cross-view localization can be enhanced by considering noisy data for weakly-supervised learning. Since aerial images have global coverage, one can access the aerial images of the target area already when training the model. In practice, even though collecting accurate ground truth data is expensive, acquiring noisy ground truth data with a positioning error of tens of meters is easy, for instance, by using phone-grade GNSS. Chapter 6 made use of this observation and proposed a knowledge self-distillation framework that utilizes this noisy data to fine-tune pre-trained cross-view localization models. The key is to select reliable pseudo ground truth generated from a teacher model to train a student model. The experiments demonstrated that the student model trained using the proposed knowledge self-distillation framework had a performance improvement of up to 20% over the direct generalization of the pre-trained model in the target area, which was previously the standard practice.

SQ5: What level of accuracy is achievable with ground-to-aerial cross-view visual localization?

The experiments conducted in Chapters 3 to 6 showed that localization accuracy significantly depends on its intended generalization, the input data, and the scene layout of the test area. These experiments explored three types of generalization: generalization to new ground images collected at different locations within the same area, generalization to new ground and aerial images from new areas, and generalization to new ground images collected on the same road but at different times. Given the absence of a single dataset encompassing all three scenarios, variations in generalization type were also associated with changes in input data, for example, panoramic images versus images with a limited FoV. In these experiments, the best-performing model, CCVPE, achieved approximately 0.5 m median lateral localization error, 0.6 m median longitudinal localization error, and a median orientation error of around 1.2° on the Oxford RobotCar dataset, which has images with a limited FoV, when generalizing to new ground-level images across time. However, when generalizing across areas on the KITTI dataset, CCVPE had a median localization error of 10.98 m.

In general, when the domain gap between the training and test sets is small, cross-view localization can achieve meter-level accuracy for localization, around 1° orientation error. However, when the domain gap is large, for example, testing in a new target area, the localization error still reaches a few meters, or even around ten meters. Even though the proposed weakly-supervised learning method in Chapter 6 managed to reduce this error by up to 20%, there is still a considerable gap when the method can be used in practice for highly automated driving in this cross-area scenario.

Notably, fine-grained cross-view localization is a new research direction established around three years ago. The accuracy of fine-grained cross-view localization has improved dramatically in this time, for example, from a median error of 7.68 m to a median error of 1.36 m on the VIGOR dataset. The author believes there is still significant room for performance improvement in this field. Currently, cross-view localization is useful for

lower-level automated driving systems, such as ADAS and lane-keeping systems, and it is likely to become highly relevant for more advanced automated driving applications in the near future.

7.2.2 Answers to the main research question:

After addressing all sub-questions, the main research question can be answered:

MQ: Can ground-to-aerial cross-view visual localization become a scalable and accurate method for estimating a vehicle's pose by comparing its captured ground-level image with an aerial image (the "map") covering its local surroundings?

Answer: Yes, ground-to-aerial cross-view visual localization can become a scalable and accurate method for estimating a vehicle's pose.

Humans are capable of localizing themselves by comparing their observed surroundings with the scene layout depicted in an aerial image. Similarly, a deep network can mimic this process when several key aspects are considered.

Firstly, ground-to-aerial cross-view visual localization should utilize the available localization prior to narrow down the search area in the aerial image. Rather than identifying the coarse location in a global context by image retrieval, cross-view localization should concentrate on fine-grained pose estimation within an aerial image patch that covers a specific local area.

Secondly, the information in ground and aerial images should be explicitly compared. For example, it is common to use a Siamese-like network, i.e., a network with two encoder branches, to separately encode the ground-level and aerial images into image descriptors, and then compare them by calculating the similarity measure. Importantly, since the content in the ground view can only be correctly matched to that in the aerial view along the correct viewing direction, deep models should also leverage the geometric relationship between ground and aerial views to embed orientation information into the image descriptors.

Thirdly, cross-view localization should consider the easily collectable data to enhance its scalability. For humans, the ability to localize themselves using an aerial image varies from person to person, and this ability is not innate. People who frequently perform this task tend to be better at it than others. Similarly, deep networks designed for this task also need to be trained with a vast amount of data. The ground truth labels do not need to detail the exact matching information between objects on the ground and their corresponding objects in the aerial image. Just as humans discern their true location, the model can learn to match the corresponding information across views by supervising only the estimated location. Once a person becomes capable of cross-view localization, their ability improves with practice. Similarly, a trained model can also enhance its performance by leveraging its own predictions through weakly- or self-supervised learning.

Although the accuracy of ground-to-aerial cross-view visual localization does not yet meet the requirements for autonomous driving, given the rapid pace of development in this field, it can become a scalable and accurate method for estimating a vehicle's pose.

7.3 Discussion

This section extends the discussion to a wider context, including the role of inductive biases in the training and designing of cross-view localization models, the use of ground truth

data as well as the discrepancies between current model capabilities and the requirements of autonomous driving.

7.3.1 Incorporating expert knowledge into cross-view localization

Recent advances in *foundation models* [92, 171, 179] try to solve various computer vision tasks at the same time with large models and broad data. Despite the impressive progress achieved in this line of research, the findings of this dissertation highlight the effectiveness of the opposite path, namely, using domain-specific knowledge to improve the learning of the task-specific deep model. This dissertation achieved this by two means, one is by learning preference from data, as demonstrated in Chapter 3 and 6, and another is by leveraging the inductive bias in the model architecture, as done in Chapter 4 and 5.

Learning preference from data: In the computer vision community, cross-view image retrieval has been considered a stand-alone localization technique, for example, as a replacement for GNSS. However, in robotics, localization is often viewed as a system-level task that necessitates the fusion of multiple sensors. Chapter 3 adopted this robotics perspective to enhance the vision task by incorporating the localization prior from GNSS into cross-view image retrieval. The loss function takes the localization prior into account to guide the cross-view image retrieval models in extracting features that are more discriminative for localization in a small local area, such as along a road. Qualitative results showed that the model trained with the localization prior often focuses on features from streetlights and poles, whereas the model trained without the prior tends to overlook these objects, concentrating more on road contours instead. Objects like streetlights and poles, despite being common in different places, are useful in disambiguating other images along the road. Road contours, while globally distinctive, lack local discriminative power. Consequently, the model trained with the localization prior outperforms the model trained without it in the local target area.

Chapter 6 introduced a knowledge self-distillation framework designed to train a student model using the predictions of a teacher model as pseudo ground truth. The framework's effectiveness relies on its method for suppressing noise and filtering outliers in the teacher model's predictions, essentially selecting preferred data for training. The experiments in Chapter 6 demonstrated that without removing the unreliable predictions of the teacher, the resulting student model could be worse than the teacher model. By eliminating undesirable pseudo ground truth, the student model significantly improves. This underscores a critical insight: simply increasing the volume of data in a model's training, without considering the quality of that data, does not guarantee improved performance on the target task.

To conclude, both Chapter 3 and Chapter 6 inject human knowledge into data selection for training a more effective cross-view localization model. Chapter 3 incorporates GNSS prior to learn a model that is more discriminative within the local target area. Chapter 6 designs methods to retain data more likely to belong to inliers, thereby improving the performance of a student model in a knowledge self-distillation pipeline. The effectiveness of the methods in Chapters 3 and 6 suggests that in practice, instead of merely adding more

data, developing methods that keep data closer to the target distribution can lead to better performance in the target area.

Leveraging inductive bias in the model: Deep models have a strong capability of learning a solution from data. However, solely learning from data is not the optimal approach, given the limited amount of labeled data and computational resources. In practice, embedding expert knowledge as an inductive bias of the model is a commonly adopted strategy to enhance model learning and generalization with limited data. There are two types of inductive biases: preference bias and restriction bias [180]. Preference bias steers the model to favor certain hypotheses over others, for example, preferring smaller weights at each layer. In contrast, restriction bias narrows down the set of hypotheses the model considers. Both Chapter 4 and Chapter 5 exploit the projection geometry to find the relation between the columns in the ground image and the rays originating from the ground truth pose in the aerial image. This geometry relation between ground and aerial images is embedded as restriction bias into the proposed methodologies to jointly consider localization and orientation estimation. As a result, both CCVPE and SliceMatch utilize the translational equivariance property of CNNs to construct an orientation-aware ground image descriptor. CCVPE uses the orientation-aware ground descriptor to guide the learning of orientation aerial descriptors. SliceMatch, on the other hand, uses the pre-computed slice masks as an additional inductive bias to force the aerial descriptor to gather orientation information. It is shown in Chapter 4 and Chapter 5 that the proposed CCVPE and SliceMatch methods achieved state-of-the-art camera pose estimation accuracy, outperforming methods that estimate location without considering orientation, such as the CVR method, by a large margin. In SliceMatch, the use of slice masks also enables an efficient architecture that runs significantly faster than other baselines.

7.3.2 Availability of data

Previous research in cross-view localization [32, 33, 127, 130, 163] often overlooked the availability and potential use of data with noisy ground truth locations, instead focusing on developing supervised learning approaches that generalize well across different scenarios. However, deploying a trained model in a new area always results in a performance drop [32, 33, 127, 130, 163]. Chapter 6 pioneered a new solution to this challenge. It was among the first to utilize data with noisy ground truth locations to adapt the trained cross-view localization models to new target areas effectively. The effectiveness of the proposed knowledge self-distillation framework was demonstrated using two state-of-the-art methods across two datasets.

Due to the challenges associated with gathering precise ground truth data on a large scale, exploring alternatives such as employing noisy ground truth in weakly-supervised learning, or adopting self-supervised learning methodologies that do not require ground truth for the training of cross-view localization models, emerges as a practical research direction. As the work proposed in Chapter 6 is the first in this direction, the author hopes that this proposed work will motivate further research aimed at enhancing the scalability of cross-view localization techniques.

7.3.3 The gap to autonomous driving requirements

As summarized in Section 1.2, the localization requirements for autonomous driving on local streets are no more than 0.29 m for both lateral and longitudinal errors and an orientation error of up to 0.50°.

Observations from experiments detailed in Chapters 3, 4, 5, and 6 show that a model's localization accuracy significantly depends on its aimed generalization, the input data, and the scene layout of the test area. The conducted experiments focus on three primary types of generalizations: generalization to new ground images collected at different locations within the same area, generalization to new ground and aerial images from new areas, such as different cities, and generalization to new ground images collected on the same road but at different times, including variations in the time of day and seasons.

The generalization to the same area and to new areas was tested using the VIGOR and KITTI datasets. Generalizing within the same area is an easier task than to new areas, given that the mode is exposed to the test scene layout, such as roads and buildings, during training. In this case, the best-performing model, CCVPE, achieved a median localization error of 1.42 m and a median orientation error of 6.62° on VIGOR, and a median localization error of 3.47 m and a median orientation error of 6.12° on KITTI. Notably, despite VIGOR's dataset being collected in US cities with typically wider roads compared to those in Karlsruhe, Germany (where KITTI was collected), the localization error on VIGOR was significantly lower than on KITTI. This discrepancy can be attributed to the ground images in VIGOR being panoramic, as opposed to KITTI's limited horizontal field of view (HFoV), making longitudinal localization more challenging. It was also observed that lateral localization errors were lower than longitudinal ones across all tested models.

Currently, consumer-level vehicles are equipped solely with front-facing cameras. With this setting, relying solely on the proposed CCVPE method still leaves a considerable gap to meeting the localization accuracy requirements for self-driving vehicles. Notably, there is a trend towards equipping vehicles with additional cameras to cover a 360° horizontal FoV. Despite the 360° view being captured by multiple images instead of a single panorama, the proposed CCVPE and SliceMatch methods can be adapted to this setting. This adaptation involves constructing a descriptor for each image separately and then concatenating all descriptors to form the final ground-level descriptor. Even though CCVPE with 360° view does not fully close the gap to meeting the localization accuracy requirements, it is important to note that fine-grained cross-view localization is a relatively new field that emerged about three years ago. Given the current pace of development and improvements in accuracy, it remains a highly relevant research direction for self-driving vehicles.

When generalized to new areas, CCVPE achieved a median localization error of 1.89 m on the VIGOR dataset and 10.98 m on KITTI. Despite the VIGOR training and test data being collected in different cities, and KITTI's data spanning different roads within the same city, the performance gap between same-area testing and cross-area testing on KITTI (a gap of 7.24 m) was still larger than that on VIGOR (a gap of 0.47 m). This does not suggest a minor domain gap between different cities in the VIGOR dataset. Instead, it emphasizes that the KITTI training set, with its densely sampled images from a few roads, leads to the model overfitting those specific roads and performing poorly in generalizing to new roads in the test set. In this setting, a significant gap exists between the autonomous driving requirement of 0.29 m for lateral and longitudinal errors and the performance

of current models. This discrepancy underscores the importance of further research for improving model performance in areas where no ground truth is available. It aligns with the author's suggestion to explore weakly- or self-supervised learning methods for cross-view localization.

The generalization to new ground images collected at different times was evaluated using the Oxford RobotCar dataset. This dissertation excludes extreme lighting condition changes, such as training with daytime images and testing with nighttime images. The primary variations considered here are related to weather and seasons, as well as dynamic objects. In this setting, CCVPE achieved a median lateral localization error of approximately 0.5 m, a median longitudinal localization error of approximately 0.6 m, and a median orientation error of about 1.2°. These results showcase the model's robust ability to adapt to static and stable objects for localization within a known area. By combining CCVPE with temporal filtering or local localization methods, such as visual odometry, there is potential to achieve the required accuracy of a maximum permitted lateral and longitudinal error of 0.29 m.

7.4 Future work

Although this dissertation has pushed ground-to-aerial cross-view localization a step forward to its real-world application for autonomous driving, there are still many challenges that remain for future research. This section first presents the possible future work in cross-view localization. Then it discusses the potential broader use of aerial images for more autonomous driving tasks. Finally, it envisions the future of environmental perception for autonomous driving.

7.4.1 Future work in cross-view localization

Based on the findings and insights derived from this dissertation, this subsection suggests possible future research in cross-view localization.

Local feature matching: In Chapter 4 and Chapter 5, both the CCVPE and SliceMatch methods infer the ground camera pose by comparing the image descriptor of the ground image with descriptors of the aerial image. For both methods, each aerial descriptor contains information from the entire aerial image.

Motivated by structure-based localization methods, which calculate camera pose based on matched local structures, e.g., salient points and corners, across different ground views, this dissertation suggests that future work in cross-view localization should also consider matching local features across ground and aerial views. Given a 3D environment, ground and aerial images represent two views capturing the scene from vastly different perspectives. Therefore, instead of matching detailed structure correspondences, cross-view methods should search for larger corresponding features, such as objects. For each object in the scene, its side is captured by the ground view, and its top by the aerial view. Establishing multiple object-level correspondences across ground and aerial views allows for calculating the camera pose based on projection geometry. Besides improving accuracy, local feature matching-based cross-view localization could potentially enhance the model's generalization to new environments, since local features are likely to be more consistent

compared to the global appearance of the image. The challenge, then, lies in acquiring accurate ground truth correspondence data to train such a model.

Pre-training: Current cross-view localization models often initialize with pre-trained weights from ImageNet or other ground-level naturalistic image datasets. Such initialization is reasonable for the ground branch in cross-view localization models but might be sub-optimal for initializing the aerial feature extractor, since the aerial view captures objects from above, and the scale of these objects is often much smaller. Aerial images are widely available all over the world, providing the possibility of using them for model pre-training. In some regions, there are also aerial images collected in different years. Pre-training from those data can potentially make the model's feature extractor adapt to the top view of objects and also potentially learn to identify dynamic objects if temporal data is used.

The main challenge, then, is designing a pre-training pipeline that can benefit the downstream localization task. Common pre-training objectives, such as self-supervised contrastive learning and masked autoencoding, might improve cross-view localization but may not be optimal. This is because cross-view localization models typically have a Siamese-like architecture, requiring interaction between two branches. Hence, an ideal pre-training scheme should consider unlabeled data from both ground and aerial domains.

Alternatively, one can leverage foundation models for cross-view localization. Recently, foundation models have been trained for applications involving satellite images [181]. Since these models are trained on vast amounts of diverse data, they can serve as more powerful feature extractors for cross-view localization models.

Weakly-supervised learning: As Chapter 6 pointed out, collecting ground-level images with coarse location estimates is relatively easy, and employing weakly supervised learning on these data can enhance the localization accuracy of cross-view localization models. The framework introduced in Chapter 6 represents one of the initial trials in this direction, utilizing knowledge self-distillation for model training. Future work could explore designing alternative weakly supervised learning objectives for cross-view localization. One example is enforcing the consistency of the output when changing the input. For instance, modifying the orientation of a ground-level panoramic image and forcing the model's orientation output to change accordingly, or shifting the input aerial image and forcing the model to output geo-location to be consistent. Training with such objectives could encourage the model to learn feature matching across ground and aerial views without accurate ground truth data.

Leveraging more ground images: So far, most cross-view localization methods match a single ground-level image to an aerial image. However, for vehicle localization, video data are often available. Therefore, cross-view localization methods should consider such data. Chapter 3 developed a particle filter-based approach that fuses per-frame outputs over time. Instead, future work should also consider directly taking temporal data as the input to the model, such that the model can better disambiguate dynamic and statics objects during matching.

Moreover, Chapters 4 and 5 found that localization in the longitudinal direction poses more challenges than in the lateral direction when only front-facing images are used. Incor-

porating side-facing images could potentially improve longitudinal localization accuracy. Nowadays, vehicles equipped with sensors for autonomous driving commonly have cameras covering a 360-degree horizontal vicinity. Therefore, future work should increasingly focus on utilizing the 360-degree surrounding view for cross-view localization. Although panoramic images are used in Chapters 4 and 5, they remain a less common image format in autonomous driving. Future work could concentrate more on using multiple ground images with a limited FoV.

More challenging scenario and corner cases Chapters 3 and 4 studied the model's generalization to ground images collected at different times, but extreme lighting or weather conditions were not included in those experiments. Future work could focus on developing cross-view matching models that are robust to challenging lighting and weather conditions, such as nighttime, heavy rain, and fog.

Besides, in self-driving datasets, such as KITTI and Oxford RobotCar, the ego-vehicle always drives on the road. This raises a concern that cross-view localization models trained on these datasets might learn a bias, assuming the vehicle's location is always on the road, regardless of where the ground-level images were taken. This becomes safety-critical if the vehicle is not on the road, but the cross-view localization model still predicts a location on the road. To address these corner cases, it is necessary to collect a dataset covering such scenarios. Subsequent research should then concentrate on developing methods that generalize well to these situations.

Variations in aerial images Current research in cross-view localization treats aerial images as the reference map but does not explore how factors such as the spatial resolution of aerial images, their size, and the time of collection influence localization accuracy. Moreover, aerial images used in cross-view localization benchmarks typically have a fixed spatial resolution. A higher spatial resolution means that each pixel represents a smaller ground area, which could potentially improve localization accuracy. However, if the size of the aerial image remains constant while its spatial resolution increases, the ground area covered by the aerial image decreases. Consequently, objects visible from the ground view might not be visible in the aerial view, negatively affecting cross-view localization accuracy. Additionally, it is crucial to acknowledge that aerial images and ground images are not collected simultaneously. The "freshness" of the aerial images can also impact the accuracy of cross-view matching.

These subtle factors could significantly influence cross-view localization accuracy. Therefore, a systematic study on these aspects is highly valuable for future research.

Adversarial attacks Cross-view localization techniques increase the risk of exposing individuals' precise location information. For instance, mobile phone images, like those from iPhones, often include GNSS geo-tags in their metadata. This approximate location can be used to identify a local aerial image patch, allowing fine-grained cross-view localization methods to pinpoint the exact location where the image was captured. Consequently, hackers could exploit this method to track individuals, such as social media influencers, by accessing the images they share online. This raises significant security and privacy concerns. To counter these risks, future research should investigate adversarial attacks [182–184] on

cross-view localization, developing methods that make negligible changes to the image but can disrupt cross-view localization methods.

7.4.2 The broader use of aerial images for autonomous driving

As introduced in Chapter 1, the information provided by a pre-constructed map can significantly reduce the dependency on the online self-driving perception stack for mapping the environment. Similarly, aerial images can offer complementary information to assist the online perception system in autonomous driving. For instance, recent advances in trajectory prediction [185–187] often depend on HD maps to forecast the future trajectories of vehicles. However, compared to HD maps, aerial images offer richer appearance information beyond just road lanes and traffic signs. Incorporating aerial images for trajectory prediction can potentially provide more environmental information to the prediction models, and thus improve the accuracy. Besides, since aerial images have global coverage, using them as inputs instead of HD maps could increase the scalability of trajectory prediction methods.

7.4.3 Applying proposed techniques on other applications than autonomous driving

Ground-to-aerial cross-view visual localization shares similarities with many other types of BEV map-based localization. Besides autonomous driving, BEV map-based localization is used for many applications. A notable example is indoor localization [86, 87, 89], e.g. localizing a robot in a shopping mall given a floor plan map. Since this task involves comparing relevant features between ground-level images and the BEV floor plan, the methods proposed in this dissertation, like CCVPE and SliceMatch can potentially be applied to this task. Future research could explore the direct application of these proposed methods for indoor localization.

7.4.4 Author's wishes

Finally, the author ends this dissertation with his wishes. When the author began his Ph.D., ground-to-aerial cross-view visual localization was primarily seen as a technique for replacing GNSS for large-scale coarse localization, rather than as a precise vehicle localization method for autonomous driving. The ECCV paper [31] covered in Chapter 4 was among the first works to delve into the fine-grained localization task, demonstrating promising results in accurate cross-view localization. Since then, the computer vision research community has shown increasing interest in this research direction, leading to advancements in the accuracy of cross-view localization at every top computer vision conference. However, the attention given to this field has not yet met the author's expectations. One major reason is the lack of commercial products utilizing this technique for vehicle localization, despite research efforts that have been made by also leading tech companies such as Google and Meta [77, 132]. Therefore, the author hopes that ground-to-aerial cross-view visual localization can lead to the development of commercial products, and that this attention will, in turn, drive further advancements in academia to push the limits of the cross-view localization technique.

In broad terms, the author hopes that this dissertation can also contribute to the development of autonomous vehicles, potentially leading to a decrease in traffic accidents and an improvement in traffic flow efficiency.

Acknowledgments

While words can convey the content of this PhD dissertation, they fall short of expressing my profound gratitude to those who guided, supported, and accompanied me throughout this journey.

First and foremost, my heartfelt thanks go to my promotor, Dr. Julian F. P. Kooij. Julian, I am deeply grateful for your invaluable guidance over the past five years. You consistently provided insightful feedback on my research, and our meetings often extended beyond the scheduled time due to the depth and productivity of our discussions—I truly cherished those moments. Your mentorship has been instrumental in my development as an independent researcher. Whenever I faced a new task, whether it was supervising a student, being a TA, presenting at a project meeting, or writing a paper, you offered detailed feedback. With each subsequent encounter, you let me approach these tasks with increasing independence, helping me build both skills and confidence. This has been an invaluable learning experience. I still remember when I first joined the Intelligent Vehicles group, and you showed me a book on supervising PhD students. Today, I believe you could write one yourself, as you embody the qualities of an exceptional PhD supervisor.

I also extend my sincere gratitude to my other promoter, Prof. Dr. Dariu M. Gavrila. As the leader of the IV group, with extensive industry experience, your vision for the group and your management skills have always impressed me. During each yearly progress meeting, your insight into long-term milestones provided invaluable direction for my journey. I would also like to thank my two industrial supervisors from TomTom, Olaf Booij and Marco Manfredi. Olaf, with your vast industry experience, you often brought unique perspectives to problems that I would not have otherwise considered. The knowledge I have gained from you is unlike anything I could find in papers. Marco, your expertise in deep learning was a constant source of inspiration and sparked new research ideas. I am deeply grateful for your guidance and insights.

I also wish to thank the committee members, Prof. Dr. Guido de Croon, Dr. Victor A. Prisacariu, Dr. Martin R. Oswald, Dr. Jan van Gemert, and Prof. Dr. Jantien E. Stoter, for examining my thesis and joining me in the final part of my PhD.

Next, I would like to thank my colleagues in the IV group. I feel incredibly fortunate to be part of such an international, open-minded, and inclusive team. Prof. Dr. Riender Happee, Dr. Barys Shyrokau, Dr. Georgios Papaioannou, Dr. Ksander de Winkel, and Dr. Frank Everdij, thank you for sharing your expertise and expanding my understanding of Intelligent Vehicles. Dr. Holger Caesar, thank you for lending me your parenting book—it was a lifesaver during my first three months as a father. Dr. Jork Stapel, Dr. Mojtaba Mirakhorlo, Dr. Marko Cvetković, Dr. Raj Desai, and Dr. Tugrul Irmak, Vishrut Jain, Varun Kotian, Chrysovalanto Messiou, Ronald Ensing, Dr. Mario Garzon Oviedo, Dr. Oscar de Groot, Dr. Thomas Hehn, and Jetze Schuurmans, I enjoyed all the conversations we had during the lunches and all the group parties. Ted de Vries Lentsch, it has been a pleasure to work with you, from your MSc thesis to your own PhD. And Mubariz Zaffar, my fellow

Localization Guy, I cherished our productive discussions in the late evening in the office. To my office mates, Dr. Ewoud Pool, Dr. Andras Palffy, Dr. Wilbert Tabone, Hidde Boekema, Alberto Bertipaglia, and Dr. Xiaolin He, it was a pleasure to share an office with you all. You were the reason I looked forward to coming to the office, even during the post-COVID when we were encouraged to work from home. Of course, to my fellow Chinese colleagues, thank you for your friendship. Yanggu Zheng, you welcomed me to Delft, and I will never forget the memorable medieval dinner we shared. To the Shandong Gang—Dr. Xiaolin He, Dr. Yancong Lin, and Shiming Wang, your vast knowledge of history, politics, and geography always amazed me.

I extend my thanks to my Chinese friends in the Cognitive Robotics department for organizing dinners and events and making my time there enjoyable. I also wish to thank my collaborators abroad, Dr. Yujiao Shi in China and Prof. Hongdong Li in Australia, for our enriching work together. My thanks go to Decai Chen in Germany, who came to Delft to be my paranymph, and to my friends in Shenzhen, Dr. Yang Zhang, for showing me around the Netherlands and helping me with my PhD application, and Jiahao Sun, for designing the fantastic cover of this thesis.

Meanwhile, I would like to thank my current postdoc advisor, Dr. Alexandre Alahi, and my colleagues at the Visual Intelligence for Transportation (VITA) Lab at EPFL: Carol Ortega, Dr. Kirell Benzi, Dr. Charles Corbière, Dr. Kaouther Messaoud Ben Amor, Dr. Taylor Mordan, Dr. Saeed Saadatnejad, Dr. Hossein Bahari, Dr. Brian Alan Tappy-Sifringer, Mohamed Ossama Ahmed Abdelfattah, Yasamin Borhani, Lan Feng, Yang Gao, Yasaman Haghighi, Mariam Hassan, Reyhaneh Hosseininejad, Po-Chien Luan, Ahmad Rahimi, Megh Shukla, and Bastien Van Delft. I feel so lucky to have joined the VITA Lab. In a short time, I have already realized there is so much I can learn from all of you.

Finally, and most importantly, I want to thank my closest ones: my father, Canwen Xia; my mother, Xiaohong He; my grandmother, Chunlian Liu; my aunt, Haiyan He; my wife, Yuxin Song; and my son, Weichu Xia. You have been my constant support throughout this journey. To my parents, thank you for raising me and guiding me to become a better person. You have always been open-minded and supportive of my decisions. Since becoming a father myself, I now understand even more deeply how selflessly you have always worked to provide the best for me. Words cannot express the depth of my gratitude. To my grandmother and my aunt, thank you for your unwavering support. It has meant so much to me. Yuxin, I could not have completed this PhD without you. When I felt discouraged or upset, you offered support and understanding, helping me regain my motivation. You know better than anyone what I have been through on this journey, and you have been the most important person helping me through it. I understand how much you have sacrificed. Being 13,000 km apart, we each spent countless hours on flights and many late nights to overcome the long distance and time difference. Since our son, Weichu, was born two years ago, you and Weichu have been, and always will be, my strongest support in facing any challenge.

Bibliography

References

[1] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *International Journal of Robotics Research*, 36(1):3–15, 2017.

[2] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-Net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

[3] Rong Liu, Jinling Wang, and Bingqi Zhang. High definition map for automated driving: Overview and analysis. *The Journal of Navigation*, 73(2):324–341, 2020.

[4] Wei-Chiu Ma, Ignacio Tartavull, Ioan Andrei Bârsan, Shenlong Wang, Min Bai, Gellert Mattyus, Namdar Homayounfar, Shrinidhi Kowshika Lakshmikanth, Andrei Pokrovsky, and Raquel Urtasun. Exploiting sparse semantic hd maps for self-driving vehicle localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5304–5311, 2019.

[5] Hao Cai, Zhaozheng Hu, Gang Huang, Dunyao Zhu, and Xiaocong Su. Integration of GPS, monocular vision, and high definition (HD) map for accurate vehicle localization. *Sensors*, 18(10):3270, 2018.

[6] David Pannen, Martin Liebner, Wolfgang Hempel, and Wolfram Burgard. How to keep hd maps for automated driving up to date. In *IEEE International Conference on Robotics and Automation*, pages 2288–2294, 2020.

[7] Zhibin Bao, Sabir Hossain, Haoxiang Lang, and Xianke Lin. A review of high-definition map creation methods for autonomous driving. *Engineering Applications of Artificial Intelligence*, 122:106125, 2023.

[8] Andi Zang, Runsheng Xu, Goce Trajcevski, and Fan Zhou. Data issues in high-definition maps furniture–a survey. *ACM Transactions on Spatial Algorithms and Systems*, 10(1):1–37, 2024.

[9] World Health Organization. *Global status report on road safety 2018*. World Health Organization, 2018.

[10] Richard S Wallace, Anthony Stentz, Charles E Thorpe, Hans P Moravec, William Whittaker, and Takeo Kanade. First results in robot road-following. In *International Joint Conference on Artificial Intelligence*, pages 1089–1095, 1985.

[11] Takeo Kanade, Chuck Thorpe, and William Whittaker. Autonomous land vehicle project at cmu. In *Proceedings of the ACM Conference on Computer Science*, pages 71–80, 1986.

[12] Sae International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE Int.*, 4970(724):1–5, 2018.

[13] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[14] Tyler GR Reid, Sarah E Houts, Robert Cammarata, Graham Mills, Siddharth Agarwal, Ankit Vora, and Gaurav Pandey. Localization requirements for autonomous vehicles. *SAE International Journal of Connected and Automated Vehicles*, 2(12-02-03-0012):173–190, 2019.

[15] Sampo Kuutti, Saber Fallah, Konstantinos Katsaros, Mehrdad Dianati, Francis Mccullough, and Alexandros Mouzakitis. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*, 5(2):829–846, 2018.

[16] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems*, volume 4, page 1, 2007.

[17] Lijun Wei, Cindy Cappelle, Yassine Ruichek, and Frédérick Zann. Intelligent vehicle localization in urban environments using ekf-based visual odometry and gps fusion. *IFAC Proceedings Volumes*, 44(1):13776–13781, 2011.

[18] B LOUIS Decker. World geodetic system 1984. *Defense Mapping Agency Aerospace Center St Louis Afs Mo*, 1986.

[19] Lambert Wanninger. Introduction to network rtk. *IAG Working Group*, 4(1):2003–2007, 2004.

[20] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 2013.

[21] Malek Karaim, Mohamed Elsheikh, Aboelmagd Noureldin, and RB Rustamov. Gnss error sources. *Multifunctional Operation and Application of GPS*, pages 69–85, 2018.

[22] Seyed Majid Azimi, Peter Fischer, Marco Körner, and Peter Reinartz. Aerial lanenet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(5):2920–2938, 2018.

[23] Yiyang Zhou, Yuichi Takeda, Masayoshi Tomizuka, and Wei Zhan. Automatic construction of lane-level hd maps for urban scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6649–6656, 2021.

[24] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9:1–11, 2017.

[25] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[26] Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, and Frédérick Carrel. A comprehensive survey of visual slam algorithms. *Robotics*, 11(1):24, 2022.

[27] Jun Cheng, Liyan Zhang, Qihong Chen, Xinrong Hu, and Jingcao Cai. A review of visual slam methods for autonomous driving vehicles. *Engineering Applications of Artificial Intelligence*, 114:104992, 2022.

[28] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *IEEE International Conference on Robotics and Automation*, pages 4628–4634, 2022.

[29] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian F. P. Kooij. Geographically local representation learning with a spatial prior for visual localization. In *European Conference on Computer Vision Workshops*, pages 557–573. Springer, 2020.

[30] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian F. P. Kooij. Cross-view matching for vehicle localization by learning geographically local representations. *IEEE Robotics and Automation Letters*, 6(3):5921–5928, 2021.

[31] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian F. P. Kooij. Visual cross-view metric localization with dense uncertainty estimates. In *European Conference on Computer Vision*, pages 90–106. Springer, 2022.

[32] Zimin Xia, Olaf Booij, and Julian F. P. Kooij. Convolutional cross-view pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3813–3831, 2024.

[33] Ted Lentsch, Zimin Xia, Holger Caesar, and Julian F. P. Kooij. Slicematch: Geometry-guided aggregation for cross-view pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17225–17234, 2023.

[34] Zimin Xia, Yujiao Shi, Hongdong Li, and Julian FP Kooij. Adapting fine-grained cross-view localization to areas without fine ground truth. In *European Conference on Computer Vision*, pages 397–415. Springer, 2024.

[35] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2938–2946, 2015.

[36] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *IEEE International Conference on Robotics and Automation*, pages 4762–4769, 2016.

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

[38] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017.

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

[40] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *IEEE International Conference on Robotics and Automation*, pages 6939–6946, 2018.

[41] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018.

[42] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018.

[43] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

[44] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[45] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Coordinet: uncertainty-aware pose regressor for reliable vehicle localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2229–2238, 2022.

[46] Sijie Wang, Qiyu Kang, Rui She, Wee Peng Tay, Andreas Hartmannsgruber, and Diego Navarro Navarro. Robustloc: Robust camera pose regression in challenging driving environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6209–6216, 2023.

[47] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3302–3312, 2019.

[48] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[49] Sivic and Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1470–1477, 2003.

[50] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[51] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010.

[52] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015.

[53] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2011.

[54] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3384–3391, 2010.

[55] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[56] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4297–4304, 2015.

[57] Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters*, 92:89–95, 2017.

[58] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place

recognition. In *IEEE International Conference on Robotics and Automation*, pages 3223–3230, 2017.

[59] Zetao Chen, Fabiola Maffra, Inkyu Sa, and Margarita Chli. Only look once, mining distinctive landmarks from convnet for visual place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 9–16, 2017.

[60] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Robotics: Science and Systems*, pages 1–10, 2015.

[61] Peer Neubert and Peter Protzel. Beyond holistic descriptors, keypoints, and fixed patches: Multiscale superpixel grids for place recognition in changing environments. *IEEE Robotics and Automation Letters*, 1(1):484–491, 2016.

[62] Zhe Xin, Yinghao Cai, Tao Lu, Xiaoxia Xing, Shaojun Cai, Jixiang Zhang, Yiping Yang, and Yanqing Wang. Localizing discriminative visual landmarks for place recognition. In *IEEE International Conference on Robotics and Automation*, pages 5979–5985, 2019.

[63] Philippe Weinzaepfel, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. Visual localization by learning objects-of-interest dense match regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5634–5643, 2019.

[64] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2136–2145, 2017.

[65] Tayyab Naseer, Gabriel L Oliveira, Thomas Brox, and Wolfram Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *IEEE International Conference on Robotics and Automation*, pages 2614–2620, 2017.

[66] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

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

[68] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6896–6906, 2018.

[69] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[70] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[71] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019.

[72] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020.

[73] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021.

[74] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 667–674, 2011.

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

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

[77] Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Bulò, Richard Newcombe, Peter Kontschieder, and Vasileios Balntas. Orienternet: Visual localization in 2d public maps with neural matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21632–21642, 2023.

[78] Mengjie Zhou, Xieyuanli Chen, Noe Samano, Cyrill Stachniss, and Andrew Calway. Efficient localisation using images and openstreetmaps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5507–5513, 2021.

[79] Georgios Floros, Benito Van Der Zander, and Bastian Leibe. Openstreetslam: Global vehicle localization using openstreetmaps. In *IEEE International Conference on Robotics and Automation*, pages 1054–1059, 2013.

[80] Fan Yan, Olga Vysotska, and Cyrill Stachniss. Global localization on openstreetmap using 4-bit semantic descriptors. In *IEEE European Conference on Mobile Robots*, pages 1–7, 2019.

[81] Philipp Ruchti, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Localization on openstreetmap data using a 3d laser scanner. In *IEEE International Conference on Robotics and Automation*, pages 5260–5265, 2015.

[82] Huayou Wang, Changliang Xue, Yanxing Zhou, Feng Wen, and Hongbo Zhang. Visual semantic localization based on hd map for autonomous vehicles in urban scenarios. In *IEEE International Conference on Robotics and Automation*, pages 11255–11261, 2021.

[83] Chengcheng Guo, Minjie Lin, Heyang Guo, Pengpeng Liang, and Erkang Cheng. Coarse-to-fine semantic localization with hd map for autonomous driving in structural scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1146–1153, 2021.

[84] Farouk Ghallabi, Fawzi Nashashibi, Ghayath El-Haj-Shhade, and Marie-Anne Mittet. Lidar-based lane marking detection for vehicle positioning in an hd map. In *IEEE International Conference on Intelligent Transportation Systems*, pages 2209–2214, 2018.

[85] Farouk Ghallabi, Ghayath El-Haj-Shhade, Marie-Anne Mittet, and Fawzi Nashashibi. Lidar-based road signs detection for vehicle localization in an hd map. In *IEEE Intelligent Vehicles Symposium*, pages 1484–1490, 2019.

[86] Henry Howard-Jenkins, Jose-Raul Ruiz-Sarmiento, and Victor Adrian Prisacariu. LaLaLoc: Latent layout localisation in dynamic, unvisited environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10107–10116, 2021.

[87] Henry Howard-Jenkins and Victor Adrian Prisacariu. LaLaLoc++: Global floor plan comprehension for layout localisation in unvisited environments. In *European Conference on Computer Vision*, pages 693–709. Springer, 2022.

[88] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[89] Zhixiang Min, Naji Khosravan, Zachary Bessinger, Manjunath Narayana, Sing Bing Kang, Enrique Dunn, and Ivaylo Boyadzhiev. LASER: Latent space rendering for 2d visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11122–11131, 2022.

[90] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[91] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computa-*

tional Linguistics: Human Language Technologies, pages 4171–4186. Association for Computational Linguistics, June 2019.

[92] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, October 2023.

[93] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[94] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[95] Shraman Pramanick, Ewa M Nowara, Joshua Gleason, Carlos D Castillo, and Rama Chellappa. Where in the world is this image? transformer-based geo-localization in the wild. In *European Conference on Computer Vision*, pages 196–215. Springer, 2022.

[96] Lukas Haas, Silas Alberti, and Michal Skreta. Pigeon: Predicting image geolocations. *arXiv preprint arXiv:2307.05845*, 2023.

[97] Brandon Clark, Alec Kerrigan, Parth Parag Kulkarni, Vicente Vivanco Cepeda, and Mubarak Shah. Where we are and what we're looking at: Query based worldwide image geo-localization using hierarchies and scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23182–23190, 2023.

[98] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[99] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps. In *European Conference on Computer Vision*, pages 536–551, 2018.

[100] Mayank Bansal, Harpreet S. Sawhney, Hui Cheng, and Kostas Daniilidis. Geo-localization of street views with aerial image databases. In *Proceedings of ACM Multimedia*, MM '11, page 1125–1128, 2011.

[101] Mayank Bansal, Kostas Daniilidis, and Harpreet Sawhney. Ultra-wide baseline facade matching for geo-localization. In Andrea Fusiello, Vittorio Murino, and Rita Cucchiara, editors, *European Conference on Computer Vision Workshops*, pages 175–186, 2012.

[102] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5007–5015, 2015.

[103] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3961–3969, 2015.

[104] Scott Workman and Nathan Jacobs. On the location dependence of convolutional neural network features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 70–78, 2015.

[105] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018.

[106] Yujiao Shi, Liu Liu, X. Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. In *Advances in Neural Information Processing Systems*, pages 10090–10100, 2019.

[107] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5624–5633, 2019.

[108] Krishna Regmi and Mubarak Shah. Bridging the domain gap for ground-to-aerial image matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 470–479, 2019.

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

[110] Yujiao Shi, Xin Yu, Liu Liu, Dylan Campbell, Piotr Koniusz, and Hongdong Li. Accurate 3-DoF camera geo-localization via ground-to-satellite image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[111] Songlian Li, Zhigang Tu, Yujin Chen, and Tan Yu. Multi-scale attention encoder for street-to-aerial image geo-localization. *CAAI Transactions on Intelligence Technology*, 2022.

[112] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3501–3510, 2018.

[113] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. Optimal feature transport for cross-view image geo-localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11990–11997, 2020.

[114] Royston Rodrigues and Masahiro Tani. Global assists local: Effective aerial representations for field of view constrained image geo-localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3871–3879, 2022.

[115] Tingyu Wang, Zhedong Zheng, Chenggang Yan, Jiyong Zhang, Yaoqi Sun, Bolun Zheng, and Yi Yang. Each part matters: Local patterns facilitate cross-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(2):867–879, 2021.

[116] Yujiao Shi, Xin Yu, Shan Wang, and Hongdong Li. CVLNet: Cross-view semantic correspondence learning for video-based camera localization. In *Asian Conference on Computer Vision*, pages 652–669, 2022.

[117] Hongji Yang, Xiufan Lu, and Yingying Zhu. Cross-view geo-localization with layer-to-layer transformer. In *Advances in Neural Information Processing Systems*, pages 29009–29020, 2021.

[118] Sijie Zhu, Mubarak Shah, and Chen Chen. TransGeo: Transformer is all you need for cross-view image geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1162–1171, 2022.

[119] Nam N Vo and James Hays. Localizing and orienting street views using overhead imagery. In *European Conference on Computer Vision*, pages 494–509. Springer, 2016.

[120] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–875, 2017.

[121] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am I looking at? joint location and orientation estimation by cross-view matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4064–4072, 2020.

[122] Sijie Zhu, Taojiannan Yang, and Chen Chen. Revisiting street-to-aerial view image geo-localization and orientation estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 756–765, 2021.

[123] Sixing Hu and Gim Hee Lee. Image-based geo-localization using satellite imagery. *International Journal of Computer Vision*, pages 1–15, 2019.

[124] Lena M Downes, Dong-Ki Kim, Ted J Steiner, and Jonathan P How. City-wide street-to-satellite image geolocalization of a mobile ground agent. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 11102–11108, 2022.

[125] Lena M. Downes, Ted J. Steiner, Rebecca L. Russell, and Jonathan P. How. Wide-area geolocalization with a limited field of view camera in challenging urban environments. In *IEEE International Conference on Robotics and Automation*, pages 10594–10600, 2023.

[126] Daniel Wilson, Xiaohan Zhang, Waqas Sultani, and Safwan Wshah. Image and object geo-localization. *International Journal of Computer Vision*, pages 1–43, 2023.

[127] Sijie Zhu, Taojiannan Yang, and Chen Chen. VIGOR: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2021.

[128] Yuxuan Hou, Yi Yang, Junbo Wang, and Mengyin Fu. Road extraction assisted offset regression method in cross-view image-based geo-localization. In *IEEE International Conference on Intelligent Transportation Systems*, pages 2934–2940, 2022.

[129] Wenmiao Hu, Yichen Zhang, Yuxuan Liang, Yifang Yin, Andrei Georgescu, An Tran, Hannes Kruppa, See-Kiong Ng, and Roger Zimmermann. Beyond geo-localization: Fine-grained orientation of street-view images by cross-view matching with satellite imagery. In *Proceedings of ACM Multimedia*, pages 6155–6164, 2022.

[130] Yujiao Shi and Hongdong Li. Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17010–17020, 2022.

[131] Xiaolong Wang, Runsen Xu, Zhuofan Cui, Zeyu Wan, and Yu Zhang. Fine-grained cross-view geo-localization using a correlation-aware homography estimator. *Advances in Neural Information Processing Systems*, 36, 2024.

[132] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. Snap: Self-supervised neural maps for visual positioning and semantic understanding. *Advances in Neural Information Processing Systems*, 36, 2024.

[133] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[134] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelhagen. Uncertainty-aware vision-based metric cross-view geolocalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21621–21631, 2023.

[135] Shan Wang, Yanhao Zhang, Ankit Vora, Akhil Perincherry, and Hengdong Li. Satellite image based cross-view localization for autonomous vehicle. In *IEEE International Conference on Robotics and Automation*, pages 3592–3599, 2023.

[136] Tim Yuqing Tang, Daniele De Martini, Dan Barnes, and Paul Newman. Rsl-net: Localising in satellite images from a radar on the ground. *IEEE Robotics and Automation Letters*, 5(2):1087–1094, 2020.

[137] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[138] Tim Y Tang, Daniele De Martini, Shangzhe Wu, and Paul Newman. Self-supervised learning for using overhead imagery as maps in outdoor range sensor localization. *International Journal of Robotics Research*, 40(12-14):1488–1509, 2021.

[139] Tim Y Tang, Daniele De Martini, and Paul Newman. Get to the point: Learning lidar place recognition and metric localisation using overhead imagery. *Robotics: Science and Systems*, 2021.

[140] Boaz Ben-Moshe, Elazar Elkin, Harel Levi, and Ayal Weissman. Improving accuracy of GNSS devices in urban canyons. In *Canadian Conference on Computational Geometry*, pages 511–515, 2011.

[141] Elliott Kaplan and Christopher Hegarty. *Understanding GPS: principles and applications*. Artech house, 2005.

[142] Lionel Heng, Benjamin Choi, Zhaopeng Cui, Marcel Geppert, Sixing Hu, Benson Kuan, Peidong Liu, Rang Nguyen, Ye Chuan Yeo, Andreas Geiger, Gim Hee Lee, Marc Pollefeys, and Torsten Sattler. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In *International Conference on Robotics and Automation*, pages 4695–4702, 2019.

[143] Julieta Martinez, Sasha Doubov, Jack Fan, Ioan Andrei Bârsan, Shenlong Wang, Gellért Máttyus, and Raquel Urtasun. Pit30m: A benchmark for global localization in the age of self-driving cars. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4477–4484, 2020.

[144] S. Zair, S. L. Hégarat-Mascle, and E. Seignez. Coupling outlier detection with particle filter for gps-based localization. In *IEEE International Conference on Intelligent Transportation Systems*, pages 2518–2524, 2015.

[145] Will Maddern, Geoffrey Pascoe, Matthew Gadd, Dan Barnes, Brian Yeomans, and Paul Newman. Real-time kinematic ground truth for the Oxford RobotCar dataset. *arXiv preprint: 2002.10152*, 2020.

[146] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[147] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014.

[148] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pages 10727–10737, 2018.

[149] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[150] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[151] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015.

[152] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[153] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[154] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[155] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Proceedings of the London Mathematical Society*, pages 666–704, 1781.

[156] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[157] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.

[158] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066, 2013.

[159] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[160] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[161] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

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

[163] Yujiao Shi, Fei Wu, Akhil Perincherry, Ankit Vora, and Hongdong Li. Boosting 3-dof ground-to-satellite camera localization accuracy via geometry-guided cross-view transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21516–21526, 2023.

[164] Siddharth Agarwal, Ankit Vora, Gaurav Pandey, Wayne Williams, Helen Kourous, and James McBride. Ford multi-av seasonal dataset. *International Journal of Robotics Research*, 39(12):1367–1376, 2020.

[165] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems*, 2021.

[166] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[167] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[168] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in Neural Information Processing Systems*, 17, 2004.

[169] Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4388–4403, 2021.

[170] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in Neural Information Processing Systems*, 32, 2019.

[171] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[172] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.

[173] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[174] Mingi Ji, Seungjae Shin, Seunghyun Hwang, Gibeom Park, and Il-Chul Moon. Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10664–10673, 2021.

[175] Qianyu Zhou, Zhengyang Feng, Qiqi Gu, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Uncertainty-aware consistency regularization for cross-domain semantic segmentation. *Computer Vision and Image Understanding*, 221:103448, 2022.

[176] Yuxi Wang, Junran Peng, and ZhaoXiang Zhang. Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9092–9101, 2021.

[177] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7640–7650, 2023.

[178] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[179] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[180] Luca Silvester Rendsburg. *Inductive Bias in Machine Learning*. PhD thesis, Universität Tübingen, 2023.

[181] Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. Geochat: Grounded large vision-language model for remote sensing. *arXiv preprint arXiv:2311.15826*, 2023.

[182] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[183] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[184] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360, 2020.

[185] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pages 895–904. PMLR, 2021.

[186] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021.

[187] Hidde JH Boekema, Bruno KW Martens, Julian F. P. Kooij, and Dariu M Gavrila. Multi-class trajectory prediction in urban traffic using the view-of-delft prediction dataset. *IEEE Robotics and Automation Letters*, 2024.

Acronyms

ADAS Advanced Driver-Assistance Systems.

BEV Bird's Eye View.

CCVPE Convolutional Cross-View Pose Estimation.

DoF Degrees of Freedom.

FoV Field of View.

FPS Frames Per Second.

GNSS Global Navigation Satellite System.

GPS Global Positioning System.

HD map High-Definition map.

IMU Inertial Measurement Unit.

LiDAR Light Detection And Ranging sensors.

RTK Real-Time Kinematic positioning.

SLAM Simultaneous Localization and Mapping.

Curriculum Vitæ

Zimin XIA

Feb 21, 1994 | Born in Changsha, Hunan, China.

Education

Oct, 2019 - Present | Ph.D. candidate at Cognitive Robotics, **Delft University of Technology**
Thesis: Ground-to-Aerial Image Matching for Vehicle Localization.

Oct, 2016 - Jul, 2019 | MSc. in Geomatics Engineering, **University of Stuttgart**

Sep, 2012 - Jun, 2016 | BSc. in Geodesy and Geomatics Engineering, **Wuhan University**

Professional Experience

Mar, 2024 - Present | PostDoc at École Polytechnique Fédérale de Lausanne (EPFL)

Oct, 2018 - Jun, 2019 | Intern & MSc. thesis student at Carl Zeiss AG, Oberkochen, Germany
Stereo Reconstruction of Human Faces with Deep Learning

List of Publications

1. **Zimin Xia**, Olaf Booij, Marco Manfredi, and Julian F. P. Kooij, "Geographically Local Representation Learning with a Spatial Prior for Visual Localization," European Conference on Computer Vision 2020 Workshops, pp. 557-573, 2020.
   Author contributions: Zimin Xia proposed and implemented the approach, conducted aerial image fetching, performed the experiments, and took the lead in writing and presenting. Marco Manfredi, Olaf Booij, and Julian F.P. Kooij contributed to the writing and provided guidance and supervision.

2. **Zimin Xia**, Olaf Booij, Marco Manfredi, and Julian F. P. Kooij, "Cross-View Matching for Vehicle Localization by Learning Geographically Local Representations," IEEE Robotics and Automation Letters, vol. 6, no.3, pp. 5921-5928, 2021.
   Author contributions: Zimin Xia proposed and implemented the approach, conducted the experiments, and took the lead in writing. Marco Manfredi, Olaf Booij, and Julian F.P. Kooij contributed to the writing and provided guidance and supervision.

3. **Zimin Xia**, Olaf Booij, Marco Manfredi, and Julian F. P. Kooij, "Visual cross-view metric localization with dense uncertainty estimates," European Conference on Computer Vision 2022, pp. 90-106, 2022.
   Author contributions: Zimin Xia devised and implemented the approach, performed the experiments, and took the lead in writing and presenting. Marco Manfredi, Olaf Booij, and Julian F.P. Kooij contributed to the writing and provided guidance and supervision.

4. **Zimin Xia**, Olaf Booij, and Julian F. P. Kooij, "Convolutional Cross-View Pose Estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 5, pp. 3813-3831, 2024.
   Author contributions: Zimin Xia proposed and implemented the approach, performed the experiments, and took the lead in writing. Olaf Booij and Julian F.P. Kooij contributed to the writing and provided guidance and supervision.

5. Ted de Vries Lentsch*, **Zimin Xia***, Holger Caesar, and Julian F. P. Kooij, "SliceMatch: Geometry-guided Aggregation for Cross-View Pose Estimation," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17225–17234, 2023. *: equal contribution.
   Author contributions: Ted de Vries Lentsch and Zimin Xia jointly devised the approach. Ted de Vries Lentsch implemented the method and performed the experiments. Zimin Xia supported in experiments and took the lead in writing and presenting. Holger Caesar contributed to the writing. Julian F.P. Kooij contributed to the writing and provided guidance and supervision.

6. **Zimin Xia**, Yujiao Shi, Hongdong Li, and Julian F. P. Kooij, "Adapting Fine-grained Cross-view Localization to Areas without Ground Truth," European Conference on Computer Vision, 2024.
   Author contributions: Zimin Xia proposed and implemented the approach, performed the experiments, and took the lead in writing. Yujiao Shi contributed to running additional baselines and writing. Hongdong Li contributed to proofreading. Julian F.P. Kooij contributed to the writing and provided guidance and supervision.