City-scale Cross-view Geolocalization with Generalization to Unseen Environments

by

Lena M. Downes

S.M., Massachusetts Institute of Technology (2020)

B.S., University of New Hampshire (2018)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Autonomous Systems

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

Authored by: Lena M. Downes
Department of Aeronautics and Astronautics
September 27, 2023

Certified by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by: Theodore J. Steiner, III
Principal Member of the Technical Staff, Draper
Thesis Supervisor

Certified by: Richard Linares
Associate Professor of Aeronautics and Astronautics
Committee Member

Certified by: Rebecca L. Russell
Principal Member of the Technical Staff, Draper
Committee Member

Accepted by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

City-scale Cross-view Geolocalization with Generalization to Unseen Environments

by

Lena M. Downes

Submitted to the Department of Aeronautics and Astronautics
on September 27, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Autonomous Systems

Abstract

Cross-view geolocalization, a supplement or replacement for GPS, provides an estimate of an agent's global position by matching a local ground image to an overhead satellite image. It is challenging to reliably match these two sets of images in part because they have significantly different viewpoints. Existing works have demonstrated geolocalization in constrained scenarios over small areas using panoramic cameras, yielding methods that have limited generalization to unseen environments or conditions and that do not quantify uncertainty. This thesis details Wide-Area Geolocalization (WAG) and Restricted FOV Wide-Area Geolocalization (ReWAG) that combine a neural network with a particle filter to achieve global position estimates for a moving agent in a GPS-denied environment while scaling efficiently to city-sized regions in unseen environments and working with either panoramic or non-panoramic cameras. One contribution is a trinomial loss function that enables accurate and computation-efficient localization across city-scale search areas of nearly 300 km^2 in size by improving image retrieval on the off-center image pairs that result from a coarsely discretized satellite image database. Another contribution is a computationally efficient method to incorporate pose information with input image pairs, which improves localization accuracy with non-panoramic cameras and off-center ground images. An additional contribution is the GKL uncertainty measure for localization outputs, which enables detection of particle filter false convergence through characterization of the particle distribution. The final contribution is a demonstration of ReWAG's ability to generalize across different times of day, seasons, weather, and cameras on data collected from a moving car in Cambridge, Massachusetts, as well as the public release of a challenging imagery dataset collected on this vehicle platform. WAG and ReWAG localize from over 1 km to less than 100 m of localization error while performing particle filter updates with less than 1% of the computation required for previous approaches.

Thesis Supervisor: Jonathan P. How
Title: R. C. Maclaurin Professor of Aeronautics and Astronautics

Thesis Supervisor: Theodore J. Steiner, III
Title: Principal Member of the Technical Staff, Draper

Acknowledgments

I have to thank my PhD advisors for their support and dedication to advising me over the past years. To Dr. Ted Steiner, I would not be here today if it weren't for you. Your pep talks to boost my confidence have done as much for my graduate experience as our technical meetings about research problems. You have poured so much effort into mentoring me, you have truly gone above and beyond what is required for a Draper Scholar advisor. Thank you so much for being a mentor and friend just as much as you have been a research advisor. To Prof. Jon How, thank you for pushing me. I will never look back and think that I could have achieved so much more if only someone would have believed that I could do more. I will never question if I reached my full potential in graduate school. I know I did, because you challenged me to. Thank you for the time that you have spent advising me. As I've progressed in graduate school I have realized how rare it is to have an advisor who you can give a paper draft to on Friday afternoon and receive back a full page of edits on Monday morning. Both of you demonstrate a level of dedication to advising that is extremely admirable, and you have made me a better researcher through it.

Thank you to the rest of my committee members, Dr. Rebecca Russell at Draper and Prof. Richard Linares at MIT, as well as my thesis readers, Dr. Dong-Ki Kim at LG and Dr. Evelyn Stump at Draper, for your time and care in providing feedback, writing papers with me, and guiding me. It has been so helpful to have so many attentive eyes on my work, both improving the work and improving my confidence.

To Carl, thank you for believing in me, supporting me, comforting me, feeding me, listening to me, distracting me, and everything else. From providing ideas on how to make better figures to suggesting ways to rephrase tricky sentences, you've done as much as you could to help me succeed. I'm so lucky to have had such a solid and stable partner by my side through this crazy journey.

To my family, thank you for your unwavering faith in me. I know that you don't understand what I do but that does not stop you from thinking that I'm that best at it. Grad school would have been so much lonelier if I had been far away from you.

To my friends, thank you for lifting me up when I needed it, and holding my hand through the tough parts. I would not have gotten through this without you, especially Kim, Kyle, Jill, Marissa, Erin, Regina, Soumya, Golda, Sydney, Adriana and Lucy. I feel lucky to have developed and strengthened friendships with you throughout my PhD.

Thank you to my community in AeroAstro, all of the friends who I've made along the way through involvement in student groups. GAˆ3, GWAE, and GSW all have played a huge part in connecting me to a wider community at MIT and lighting my passion for connecting others.

Thank you to my labmates, whose brilliance, passion, and work ethic has been an inspiration to me. It was lovely to have a lab environment full of busy people working on really cool research who are still willing to drop what they are doing to help others and share their knowledge.

I also have a series of acknowledgments to make to the powerful female mentors who I have to thank for the support and guidance that has carved my academic life. Firstly, my high school physics teacher and science team coach Nancy Donahue and my high school biology teacher and science team coach Maria Annunziato. Both of you championed me and provided me endless encouragement to pursue STEM. I did not realize that I was especially good at science before your interference and insistence that I take AP physics and join science team. You set me on my path to become an engineer. Next, I would like to thank my mentor, undergraduate senior project advisor, kickboxing instructor and friend throughout my undergraduate career, Professor May-Win Thein. May-Win, I can't say enough what a positive example you set for me as a powerful woman in engineering and in academia. Your fire and passion was contagious, and was a huge factor in my decision to major in mechanical engineering and to stick with it when things got tough. I also have to thank my manager for my first internship at Draper, Jennessa Kimball. Jennessa, you took a chance on me and offered me a software engineering internship at Draper when I had never even taken a computer science class. You spent so much time and effort meeting with me regularly, connecting me with others at Draper, and encouraging me on my decision to apply

to graduate school. You set an amazing precedent for me that women belonged in the coding world and exposed me to a field that remains my passion today.

This work was supported by the Draper Scholars program. To Draper, thank you for the generous support of my graduate research, from my master's to my PhD. Your financial support and the access you gave me to brilliant people and unique hardware has propelled me forward.

In memory of Janice Zhou: your death has drastically shaped my graduate school experience. I wish you could be here, I wish you had more time at MIT, I wish we could have had the opportunity for our friendship to develop outside of COVID isolation. I've thought of you often throughout this PhD and I've tried to make something good out of the impression you left on me.

Contents

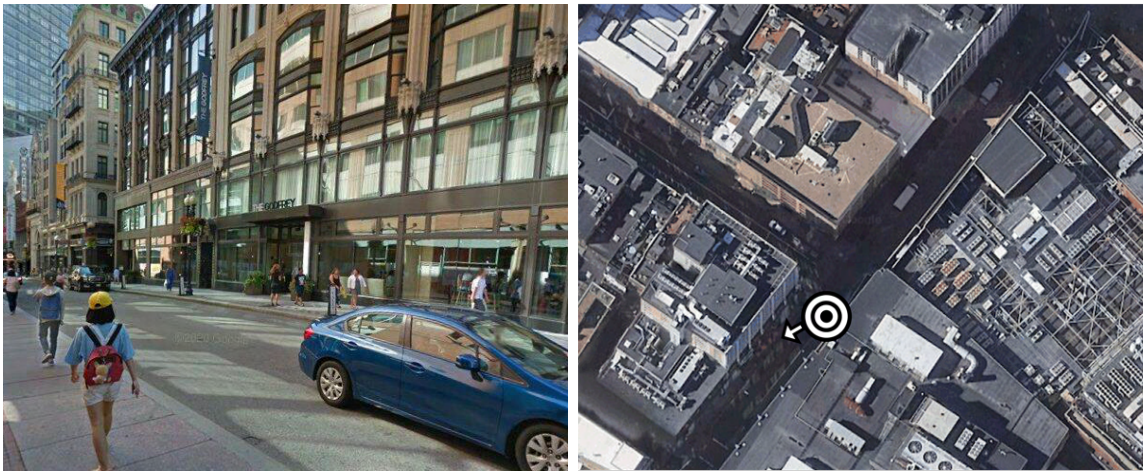
List of Figures

List of Tables

Chapter 1

Introduction

Cross-view geolocalization is the process of matching ground images to aerial images to determine the location where the ground image was taken from. Cross-view geolocalization is useful for many applications requiring location information, like autonomous vehicles, augmented reality, and ground robots. Figure 1-1 shows the challenge of this task with an example from Boston, Massachusetts. For a person familiar with Boston, it may be a trivial task to localize this ground image. For someone who has never visited Boston, matching this ground image to the aerial image may



(a) Ground view image of Downtown Crossing

(b) Aerial image of Downtown Crossing with pose of ground view camera overlaid

Figure 1-1: The visual difference in appearance between a ground view and aerial image of Downtown Crossing in Boston, Massachusetts. Imagery from Google Street View and Google Maps.

be very challenging. There is very little overlap in scene content that is visible from both perspectives; the road is visible in both images, but different parts of objects are visible in each image. In the ground view image, the facades of buildings are visible. In the aerial image, the roofs of buildings are visible. Cross-view geolocalization often does not have the advantage of having seen images of a location before, since ground view images of every location may not be available. The ultimate goal of cross-view geolocalization is to accurately match between ground and aerial imagery without previous exposure to that location and across a variety of conditions.

Matching between ground and aerial images is challenging on its own, but what happens if there are differences in season between the ground and aerial images? What if there are differences in the weather? What if there are differences in the time of day? All of these factors take the already difficult problem of matching between ground and aerial images and make it even harder. The cross-view geolocalization system must be able to focus on the invariant features across all of those changes to match between such disparate images.

Incorporating a temporal aspect to cross-view geolocalization of a ground agent as it moves improves localization by combining odometry with images over time [40, 47, 109, 117]. Satellite imagery is widely available even in areas without existing ground-view imagery, and it is the only outside data needed for cross-view geolocalization. However, the huge differences in appearance between ground and satellite images makes this problem difficult, and has caused previous systems to impose many constraints in order to make the problem tractable. In particular, preceding works had small search areas [40, 47], have relied on panoramic ground imagery [27, 40, 47, 109, 117], and have largely not provided analysis of performance under changes in time of day, weather, or distance between testing and training locations [100]. These constraints have resulted in cross-view geolocalization systems that are less accurate in many of the most interesting applications. This thesis presents a new approach to remove these constraints and create a scalable cross-view geolocalization system that can reliably provide localization information across a broad set of applications.

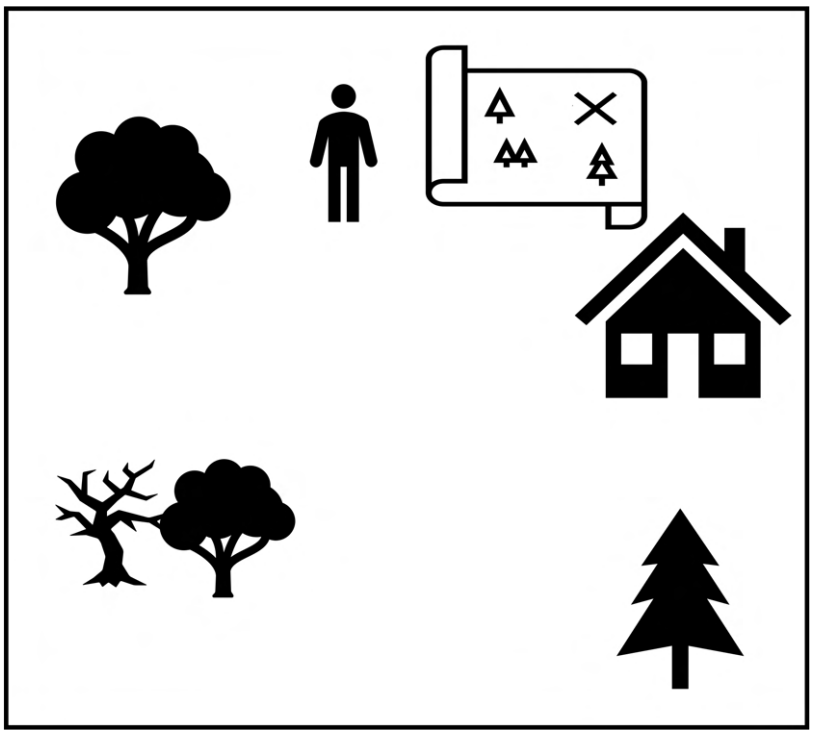


Figure 1-2: How can a lost person use a map to figure out where they are and subsequently reach a goal? They start by matching between the features in their map and the features they see.

1.1 Background

A basic function of human autonomy is knowing and understanding where one is, or localizing. Without being able to localize, a human cannot plan a path to reach a destination, cannot avoid getting stuck, cannot determine when one has reached the destination. Imagine a person falls asleep as a passenger in a car, and upon waking is dropped off in an unfamiliar location with only a map. They are then given a task, something like "go to 27 Main Street". To complete the task, they must first figure out where they are in the map. As illustrated in Fig. 1-2, the person must match between what they see around them and the features that they see in the map. This situation is an analogy for a common occurrence in an autonomous robotic system; it is powered on in a new location, unable to use its sensors until it is turned on, and given a task and a map of the environment. Localization is essential for accomplishing the task.

Although localization is essential for basic autonomy, it remains a challenge. Even

some humans find localization challenging, evinced by the smartphone-based GPS applications that have become ubiquitous in the past decade. GPS, or Global Positioning System, refers to the United States government's satellite-based navigation system, which is made up of 24 to 32 satellites in Earth orbit [31, 35]. However, GPS is just one of many constellations of satellites in the Global Navigation Satellite System (GNSS), which includes Europe's Galileo, the United States' GPS, Russia's Global'naya Navigatsionnaya Sputnikovaya Sistema (GLONASS) and China's BeiDou Navigation Satellite System (BeiDou). Although GPS is an innovation that changed the world and has provided a novel method for localization, it is not infallible; GPS is an external system that is susceptible to failure. The governments of China, India, Russia and the United States have developed anti-satellite (ASAT) technology capable of destroying GPS satellites in orbit [21]. Equipment malfunctions have led to multi-day outages [5] or hours-long degradation issues [6] of GPS systems. In addition to interference caused by human action, GPS is also vulnerable to disruption through external forces like solar storms [3].

Beyond systemic issues that could impact GPS, individuals may also be affected by problems on the receiver end like jamming [39], spoofing [95], or signal dropout due to dense foliage or urban canyons [20]. Jamming is the process of blocking GPS signals from reaching a receiver. GPS jammers can fit in the palm of a hand and can be illegally purchased for under $30. Spoofing is the process of generating false GPS signals on the same channel as legitimate GPS signals, pushing messages to the receiver that contain false information. Signal dropout is an inherent result of a system that relies on the triangulation of signals from multiple satellites. With obstructions overhead like a tree canopy or tall buildings close together, an insufficient number of satellites may be able to reach the receiver, yielding lower localization accuracy. The problem posed by dense foliage is illustrated by Fig. 1-3, where bamboo grows in such volume that the sky is barely visible. Urban canyon effects, like that seen in Fig. 1-4, can be caused by densely clustered buildings of as little as three stories [68]. In addition to intentional GPS interference from bad actors targeting a system or user, many instances of GPS interference are accidental or collateral to another intended

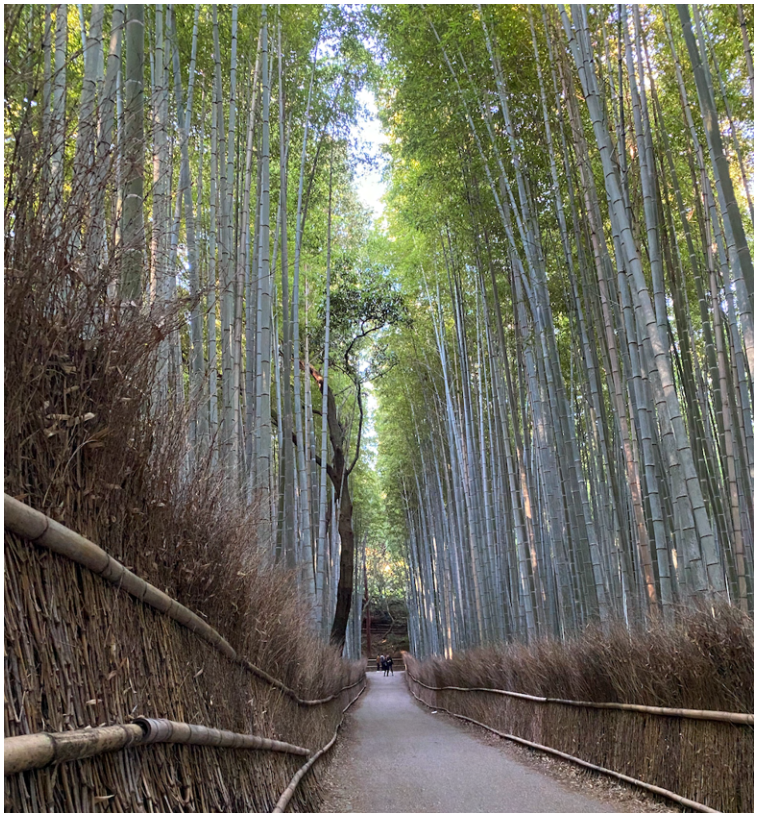


Figure 1-3: Dense foliage such as this thicket of bamboo in Arashiyama, located in Kyoto, Japan, poses a problem for GPS localization.

target. For example, in the past two decades there has been an increase in GPS-based monitoring of different types of vehicles, like freight, food delivery, and passenger transport vehicles. Many drivers resent this monitoring, and attempt to disrupt it with portable GPS jammers. These jammers disrupt the GPS localization of the user's vehicle, but it also may interfere with other satellite-based localization systems within the vicinity [78]. As a result, full reliance upon GPS for localization increases risk and creates a weakness that is not acceptable for safety-critical applications like military operations or driverless cars.

GPS-denied localization requires the processing and fusing of multiple sources of information [58]: in a suburban or urban environment this might include the geometry of street intersections [104], street names [75], the appearance of nearby buildings [85],

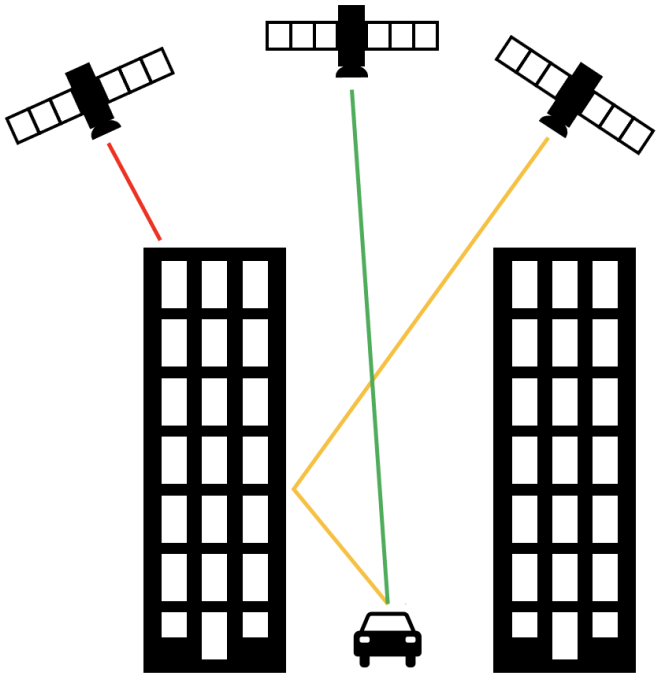


Figure 1-4: In urban canyons, multi-story buildings block GPS satellites from reaching receivers (shown with red line) or cause the signal to reflect off buildings before reaching the receiver (shown with yellow line), both of which degrade GPS accuracy.

and the existence and quantity of foliage and green spaces [62]. Vision-based localization is especially attractive due to the small size, weight, and power requirements of cameras. In addition, cameras are ubiquitous and autonomous platforms are commonly already equipped with a camera.

Satellite imagery is widely available for most of the planet through datasets like Landsat, Sentinel, MODIS, ASTER, GEOS, NOAA, Maxar and Airbus, even in forested or urban areas where it can be more challenging to use GPS. Additionally, new satellite imagery of Earth is taken every day, so imagery is frequently updated and higher resolution imagery is often made available over time. Beyond availability, satellite imagery presents lesser storage challenges than ground imagery. A satellite image of a city can be taken at a variety of resolutions, from 30 centimeters per pixel with Airbus Pléiades Neo to 30 meters per pixel with Landsat. Aerial imagery can be taken at an even higher resolution, like 15 centimeters per pixel for MassGIS imagery of Massachusetts. However, ground imagery cannot be scaled linearly in this way. Resolution of each ground image scales less favorably because there still needs

to be a ground image taken for each location where localization may be attempted. For example, the popular visual localization benchmark Oxford RobotCar Dataset has approximately one image each meter [60]. The city of Boston has approximately 934 miles of roads [4], which would equate to over one million images if it were photographed as densely as the Oxford RobotCar. In contrast, the same area could be covered in 15 cm/pixel resolution with approximately 60,000 satellite images. Hence, localization with ground imagery imposes higher storage requirements. Satellite imagery can also be more quickly updated if changes take place on the ground. Satellites can quickly be tasked to take new imagery if a natural disaster occurs or new construction is done, but refreshing ground imagery for a region would require significant time and resources. Furthermore, the use of satellite imagery as the reference imagery in practice provides a map, so simultaneous localization and mapping (SLAM) [93] does not need to be performed and the ground images just need to be localized within the given satellite map.

Cross-view geolocalization is a localization method that only requires images from a ground-view camera and preexisting overhead imagery, with no requirement for GPS measurements. Cross-view geolocalization measures the similarity between a ground image and all of the satellite images in a search area to determine the location that the ground image was taken from (see Fig. 1-5). There are many applications where cross-view geolocalization can be useful, including: robotic platforms for self-driving cars, last mile delivery, and military tactical awareness. Self-driving cars, also referred to as autonomous vehicles, may often need to drive in dense urban areas where canyon effects reduce GPS localization accuracy and may cause repeated bouncing of the estimated locations. Last mile delivery robots are small ground robots that automate the most expensive part of the shipping process, the last mile of the trip to bring a package to a customer's door, by autonomously moving packages. Similar delivery robots are also used for food delivery, a growing international market with a global market size of several billion dollars each year. The problem with these delivery robots relying solely on GPS localization is spoofing; a package or food thief could trick the robots into delivering items to a different location than the

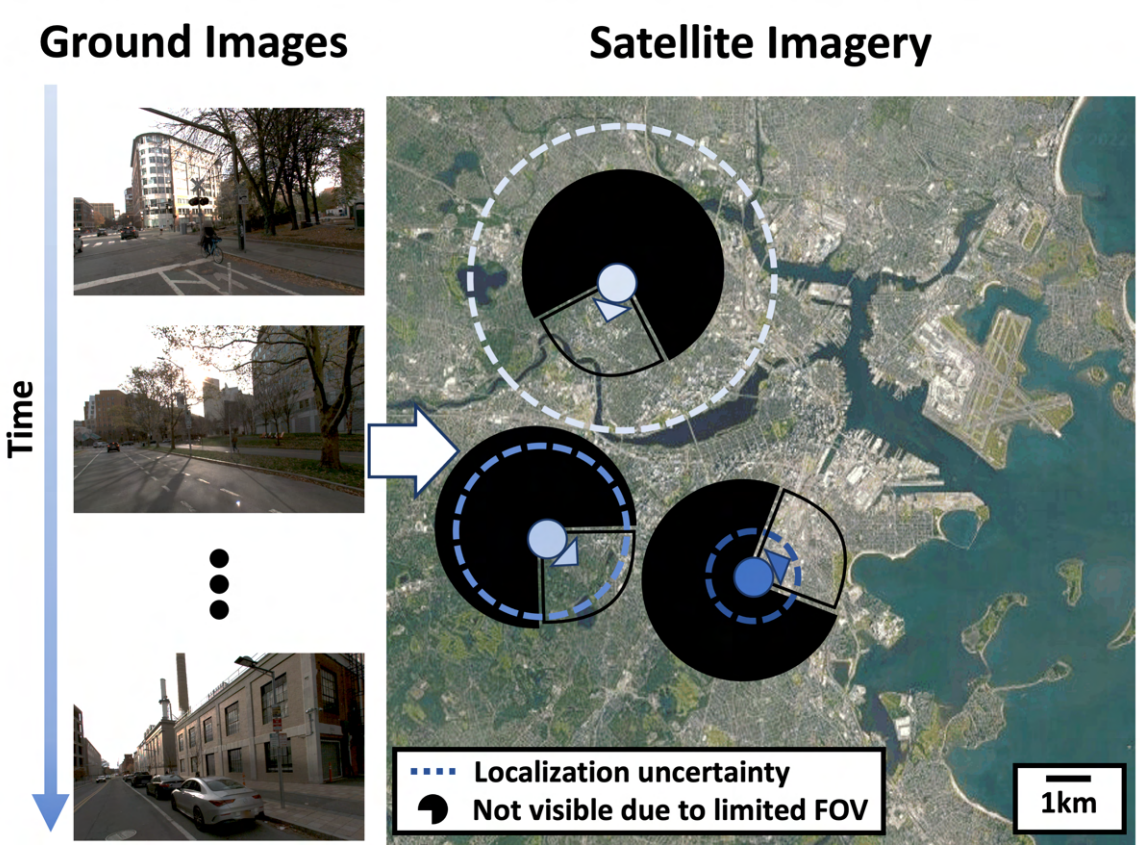


Figure 1-5: A cross-view geolocalization system takes in ground-view camera images and satellite imagery of a search area to localize an agent.

sender intended by spoofing the GPS signal. Military units often use the popular Android Tactical Assault Kit (ATAK) platform, an Android smartphone application, for geospatial and situational awareness. Military agents cannot risk the vulnerability of being GPS-dependent in an adversarial environment, and as such a cross-view geolocalization system could reduce risk and improve safety. In a military scenario, cross-view geolocalization could be used for both autonomous systems and human operators with smartphones. All three of these applications could be expected to have large search areas that are defined a priori, enabling download of relevant satellite imagery, inertial measurement units (IMUs) to obtain odometry measurements, and ground-view cameras.

Deep learning can be applied to cross-view geolocalization through the use of Siamese networks [11]. A Siamese network consists of a pair of neural networks with matching architectures that simultaneously learn embedding schemes for ground and

overhead images. The Siamese network is trained so that images taken in similar locations are close together in embedding space, and images taken at different locations are embedded far apart. The accuracy of the Siamese networks can be further improved by combining multiple measurements over time with odometry.

1.2 Motivation

Matching between query and database images is a task which increases in difficult as database size grows. A smaller database means that there is a higher likelihood to select the correct match to an input query image based on random chance alone. A larger database is also more likely to contain similar images that could introduce ambiguity in matching. Although large databases are challenging, large search areas are common in many applications of cross-view geolocalization, and large search areas typically require large satellite image databases. However, the exact size of the satellite database is determined by the way in which the search area is discretized: coarse or dense discretization. The difference between coarse and dense image discretization is illustrated in Fig. 1-6. Most existing cross-view geolocalization work has focused on small search areas with dense satellite image discretization. Dense discretization results in each part of the search area being represented in more than one database image. Dense database generation is equivalent to generating a matrix of center points and sampling a satellite image centered on each point, such that the edges of those satellite images overlap. Coarse discretization ensures that each part of the search area is represented in only one database image. Coarse database generation is equivalent to dividing the search area into a grid of satellite tiles. Existing works have used dense discretization because it simplifies the Siamese network training procedure, but coarse discretization generates small databases with even large search areas and hence reduces the difficulty of image retrieval.

Cross-view geolocalization with ground and overhead images is challenging due to the wide difference in viewpoints. Using a panoramic ground camera decreases the problem dimensionality by reducing the impact of heading. Although the alignment

(a) Coarse satellite tile

(b) Coarse satellite tile

(c) Overlapping satellite tile

(d) Overlapping satellite tile

(e) Overlapping satellite tile

(f) Full search area

Figure 1-6: Coarse satellite image sampling results in no overlap between satellite tile. Dense satellite image sampling generates overlapping satellite images and results in a larger database.

of panoramic ground images is affected by heading, the heading does not affect the content, whereas the heading of a limited field of view (FOV) camera affects the visible content of the image. The use of panoramic ground cameras also simplifies the problem by maintaining as much semantic similarity as possible between the two viewpoints – overhead images show 360° of the surroundings of a ground agent, as do panoramic ground cameras. However, in practice panoramic cameras are rarely used due to their high monetary cost, resulting in lesser availability, and their difficulty to mount without occlusion. Consequently, few real-world systems can benefit from panoramic-based localization. Widespread adoption of cross-view geolocalization technology requires its applicability to platforms without panoramic imaging capabilities.

Generalization of cross-view geolocalization systems is difficult because images of the same places can appear vastly different under difference conditions, and images of one location can contain features that are not easily learned from images of another location. Many existing works focus on improving geographic generalization, or generalizing on training images from location to testing images from another location. Although this is an important aspect of generalization, many other challenging aspects of generalization can affect cross-view geolocalization performance, like lighting conditions, seasons, time of day, and weather. Google Street View is a popular and accessible source of ground imagery for testing cross-view geolocalization since it has imagery for many locations that can be queried from a user-friendly API. Google Street View enables testing of geographic generalization but limits testing of other visual generalization tasks, since Street View images are not explicitly taken to represent a variety of visual conditions. In cross-view geolocalization applications, images have much greater visual variation than Street View images and localization performance needs to be robust to this variation.

Localization uncertainty quantification for autonomous systems is important because localization outputs are used to make navigation and planning decisions that may be safety-critical. Placing too much trust in localization outputs can result in dangerous movements, like a self-driving car crossing lanes of traffic or a delivery

robot driving across highly angled terrain instead of on flat sidewalks. Few previous cross-view geolocalization systems have attempted to quantify localization uncertainty, instead focusing on improving localization accuracy. However, as long as image matching accuracy is below 100%, uncertainty needs to be quantified to ensure safety.

1.3 Problem Statement

Overall, this thesis presents a solution to the following important problems related to cross-view geolocalization:

1. **Wide-area geolocalization.** Mobile autonomy platforms like autonomous cars, delivery robots, or military robots, can operate on longer time scales without human interaction if they are not limited to small operational areas. For cross-view geolocalization, the factor that limits the size of the operational area is its search area to localize within. The search area restricts where the ground agent can travel, because it cannot be localized if it travels outside of the search area. Hence, if a cross-view geolocalization system can only support search areas on the scale on neighborhoods, then an autonomous platform that relies upon cross-view geolocalization is limited to only operate within one neighborhood. With such a small operational area, the potential uses of these autonomous platforms is severely limited. To enable widespread use of cross-view geolocalization for many different autonomous systems, the localization system must be capable of scaling across a wide range of search area sizes.

2. **Narrow field of view ground image retrieval.** Panoramic cameras deliver rich, feature-filled images of scenes and hence could appear an attractive option for cross-view geolocalization. However, panoramic cameras present a number of challenges for autonomous platforms. For one, they are orders of magnitude more expensive than standard cameras [53], and as a result, they are less likely to be an existing component of an autonomous system. If cross-view geolocal-

ization requires the purchase and installation of an expensive sensor then the types of systems that can implement it are greatly reduced. In addition, it may not be feasible for a panoramic camera to be mounted on smaller platforms– for example, a panoramic camera might have to be mounted on top of a pedestrian's head to avoid significantly obstructing the field of view. Non-panoramic cameras are cheap, ubiquitous, and can be mounted in many ways. For these reasons, it is desirable to be able to perform accurate cross-view geolocalization with non-panoramic cameras.

3. **Generalization.** For useful deployment to realistic robotic applications, cross-view geolocalization systems need to be able to generalize to different conditions that are not seen in the training dataset. Google Street View provides a source of ground images for many locations across the world. However, Google Street View prioritizes publishing high-quality images on sunny days, and often returns to the same areas multiple times to collect the clearest imagery. During deployment of a localization system, there will be much higher variance in the conditions and quality of the ground images, since these factors are not being explicitly controlled for. For example, there may be cloudy or dark weather conditions and cameras will often be lower quality (lower resolution and narrower field of view) than those of Google. As a result, it is important for a cross-view geolocalization system to be capable of generalization to a variety of conditions that are not widely represented in training data.

4. **Localization uncertainty quantification.** A vehicle's primary navigation system fuses multiple sources of information with varying noise levels. Understanding and quantifying those noise or uncertainty levels is key to obtaining the most accurate state estimate from the navigation system. Hence, for integration into a primary navigation system, cross-view geolocalization measurements need some kind of quantification of the measurement noise or uncertainty. This uncertainty also informs the scenarios in which the proposed system can be expected to perform optimally or sufficiently, especially in the context of expanding gen-

eralization. When improving the generalizability of cross-view geolocalization systems, there are likely to be more difficult situations where localization may be uncertain. For example, if the system is being deployed during the winter with visible snow on the ground, it is important to understand whether and how much performance degrades when matching against non-winter satellite imagery. Accordingly, a cross-view geolocalization system for autonomous platforms should provide some level of uncertainty quantification.

1.4 Contributions

This thesis investigates the following main contributions to address the problems stated in Section 1.3.

1. **Coarse satellite image database and trinomial loss.** To perform localization across large, city-scale search areas, this thesis presents Wide-Area Geolocalization (WAG). WAG receives training with a novel loss function, called trinomial loss, which improves image matching performance on off-center image pairs. Improved matching performance on off-center image pairs enables localization with a coarse satellite image database since each satellite image can effectively match to a wider area of ground images. In Chapter 3, WAG is demonstrated in simulation and consistently results in more accurate position estimates, faster particle filter convergence, and lower computational and storage burdens than a baseline. This contribution addresses the problem of wide-area geolocalization and corresponds to the IROS 2022 paper [27].

2. **Pose-aware embeddings for narrow field of view and off-center image pairs.** A limitation of existing cross-view geolocalization systems is their reliance on panoramic cameras for their ground imagery. Chapter 4 presents Restricted FOV Wide-Area Geolocalization (ReWAG), a framework for generating and incorporating pose-aware embeddings to create more informative embeddings and enable accurate localization with narrow field of view cameras.

The pose provides the neural network with additional information that tells it where within the satellite image to look for matching features, which improves localization accuracy with non-panoramic cameras and with off-center ground images. This contribution addresses the problem of narrow field of view ground image retrieval and corresponds to the ICRA 2023 paper [28].

3. **Experimental demonstration of generalization with commercial off-the-shelf sensors.** Chapter 5 demonstrates ReWAG*, a generalizable version of ReWAG, with realistic, challenging data collected using commercial off-the-shelf (COTS) products. This data was collected on a moving car equipped with a camera and IMU driving in the greater Boston area. These results demonstrate the ability of this system to be practically useful without the need for specialized hardware or computation abilities, and to generalize across weather, time of day, camera, location, and season. This contribution addresses the problem of generalization and corresponds to the extension of the ICRA 2023 paper [29].

4. **Uncertainty quantification of geolocalization system.** Chapter 6 proposes GKL uncertainty, a method to generate uncertainty values for the output location estimate of ReWAG*. This uncertainty makes ReWAG* suitable for integration with a navigation system and enables more reliable performance in variable conditions, like changes in weather, time of day, or testing location. The uncertainty is measured through the estimation of the Kullback-Leibler (KL) divergence between the distribution of particles at each time step and a Gaussian distribution. GKL uncertainty can identify and quantify how multimodal a distribution is, approximating aleatoric uncertainty. GKL uncertainty better detects convergence than standard deviation, an alternate method for particle filter uncertainty quantification. This contribution addresses the problem of localization uncertainty quantification.

1.5 Summary

This chapter has provided an introduction to cross-view geolocalization, its importance, its challenges, and the contributions of this thesis. The next chapter will begin a deeper exploration into background on cross-view geolocalization and related works, which will further contextualize the contributions of this thesis.

Chapter 2

Background and Related Works

The contributions of this thesis lie at the intersection of computer vision and robotics, specifically within the application of ground-to-aerial cross-view geolocalization to mobile autonomous systems. Cross-view geolocalization is a field that has grown in popularity in recent years, in part fueled by the explosion in capabilities of machine learning for computer vision. The success of convolutional neural networks, which enable the understanding of high-level concepts from images, combined with expanding computational resources and a growing quantity of potential training images online have combined to create the perfect conditions for rapid developments in learning-based image understanding capabilities. Cross-view geolocalization lends itself perfectly to learning-based computer vision. The two main challenges to cross-view geolocalization are high-level understanding of images to compare possible image pairs for matches and the computation to search through many possible image pairs to find the matching pair. Computers are much better than humans at scaling computation, and machine learning is steadily improving computers' abilities to develop high-level understanding of images. Concurrently, concern about GPS-dependence has grown over recent years as more safety-critical systems begin to rely upon GPS. Cross-view geolocalization is an appealing technology to reduce that reliance by supplementing or replacing GPS with imagery-based localization.

The field of cross-view geolocalization includes research on matching single query images to database images, called image retrieval, and matching sequences of query

images to database images, often using probabilistic methods to do so. Cross-view geolocalization also includes research on matching images from any two perspectives, which can include drone-to-satellite or ground-to-aerial. This thesis focuses on the latter. Many works have addressed part of the problems that this thesis identifies in applying cross-view geolocalization to robotics, but none holistically address all aspects of the technical gap. This overview of related works aims to provide relevant background and to detail the strengths of existing cross-view geolocalization works while highlighting the challenges of their application to robotics.

2.1 Localization

Mobile robot localization is the process of determining a robot's position and orientation (pose) within a map of its environment. Localization is a key component to a navigation system, because a robot cannot arrive at its goal before first knowing where it is [93]. Cameras are a common sensor to apply to the localization problem due to their low cost and ubiquity; their growing use has contributed to the growth of visual place recognition.

2.1.1 Visual Place Recognition

There are many methods for using camera imagery to localize; one method that can be used for localization is visual place recognition (VPR) [59]. VPR aims to match between images of the same place that were taken at different times, like in Fig. 2-1. For example, an autonomous platform could travel around a neighborhood once to record imagery and then return a few months later, when it would use the previously-recorded imagery to match against for localization. Alternatively, VPR could be used during simultaneous localization and mapping (SLAM). SLAM is aptly named, combining localization and mapping at the same time. In SLAM, the autonomous platform does not record images of an area in advance; instead, the map is built while the agent is deployed and at the same time the agent localizes within that map that is being built. In a VPR context, SLAM would mean that the agent could

(a) A view on Memorial Drive in Cambridge, MA on a cloudy day

(b) The same view along Memorial Drive in Cambridge, MA on a sunny day

Figure 2-1: Visual place recognition matches scenes from images taken at different times.

be deployed to collect images and build a map and if it returned to a location that it had already mapped then it could use VPR to recognize its location within that map. Successfully recognizing that an agent is returning to a location that it has been to before is called a "loop closure". Loop closures which helps to correct and improve both the map and the localization in SLAM. VPR is a valuable field that has demonstrated success in many different environments. However, VPR requires matching between images taken from the same perspective, ground-view, and not all places on Earth have existing datasets of ground-view imagery.

Visual place recognition (VPR) is a field that has some overlap with cross-view geolocalization and which has existing literature discussing generalization across conditions relevant for robotics. There generally exists a tradeoff between invariance and discriminative power in VPR [96], meaning that better generalization to new environments yields lower accuracy. Nonetheless, there have been some developments in VPR that have been shown to improve generalization without harming accuracy too greatly. One technique for improved generalization is to incorporate temporal information: for example, sequences of images [23, 66, 67, 91, 96]. VPR, like cross-view geolocalization, is typically is performed as an image retrieval problem, where a single query image is compared to a database of reference images to find a match. An additional structure is necessary to localize with a sequence of images.

2.1.2 Navigation Filters

Navigation filters are one way to incorporate temporal information across a sequence of images for localization. Stochastic filtering determines the state of of a system from noisy sensor observations through recursive updating. In mobile robot navigation, filtering algorithms enable determination of the robot pose. Kalman filters [46], extended Kalman filters [79], unscented Kalman filters [103] and particle filters [22] are some common types of navigation filters. Kalman filters are applicable to linear systems, whereas extended Kalman, unscented Kalman, and particle filters are applicable to nonlinear systems and hence are more flexible. Particle filters are beneficial for cross-view geolocalization because they are not constrained to only modeling certain kinds of state distributions; particle filters can approximate any distribution. It is especially relevant for cross-view geolocalization that particle filters can approximate multimodal distributions. Perceptual aliasing can cause ground images to yield high similarity for multiple locations, which particle filters are capably of representing in the state distribution.

Particle filters use a set of samples, or particles, to represent the probability distribution of a process given noisy observations. In cross-view geolocalization, particle filters estimate the agent location given odometry information and similarity measurements from the Siamese network. Algorithm 1 shows how particle filters use random discrete particles, P_t to estimate a probabilistic distribution of a state hypothesis at time t, x_t, given control inputs, u_t, and sensor measurements, z_t.

The particle filter algorithm is made up of three main steps; propagate, weight, and resample. The propagate step uses the process model, $p\left(\mathbf{x}_t \mid u_t, \mathbf{x}_{t-1}^j\right)$, to propagate the particles. The process model consists of the control output applied to the agent or the measured odometry of the agent with added Gaussian noise. Noise is added to account for the uncertainty in how the control output affects the agent or inaccuracies in the measurement of the agent odometry.

The weight step weights the particles according to how well they match the sensor measurements using the measurement model, $p\left(z_t \mid \mathbf{x}_t^j\right)$, and normalizes the weights

Algorithm 1 Particle Filter Step for t > 0

Require: N particles at $t-1 : P_{t-1} = \left[\mathbf{x}_{t-1}^j, w_{t-1}^j\right]_{j=1\ldots N}$
  for $j = 1$ to N **do**
    sample $\mathbf{x}_t^j \sim p\left(\mathbf{x}_t \mid u_t, \mathbf{x}_{t-1}^j\right)$ $\{Propagate\}$
    compute $w_t^j = p\left(z_t \mid \mathbf{x}_t^j\right)$ $\{Weight\}$
  end for
  normalize weights w
  $\{Resample\}$
  for $j = 1$ to N **do**
    draw i with probability distribution w
    add $[\mathbf{x}_t^i, w_t^i]$ to P_t
  end for
  return P_t

to get a probability for each particle. For many problems, the correct measurement model is an open question. The most accurate measurement models require the ability to simulate what measurement z_t would be received at a specific state $\mathbf{x}_t^j$. For more simple sensors that have well modeled characteristics, like a range sensor, it is possible to build a fairly accurate measurement model. For cameras, it is not trivial to simulate the image that would be received at a specific state. Image simulation is an active field of development that often requires dedicated physics modeling of the refraction of light and the material properties of the surfaces that light is bouncing off of. Recent work has shown impressive and promising results on this task with neural radiance fields [65], but it is still challenging to generate these simulated images with limited computation [61].

The resample step replicates high probability particles and discards low probability particles. The probability of replicating a particle is proportional to its probability. Resampling helps to prevent all of the probability from concentrating in only a few particles, a problem called particle degeneracy. There are many different algorithms for resampling, including multinomial, systematic, stratified and residual resampling [38].

2.1.3 Particle Filter Uncertainty

To use cross-view geolocalization systems in real-life applications, it is necessary to know measurement covariance and how sensitive the system is to conditions that are different than the training conditions. Without knowing this it is dangerous to trust the system outputs, which could be inaccurate [45]. There are many metrics by which to measure this sensitivity. Although the similarity of the Siamese network embeddings is often used as a pseudo-measurement of match confidence, previous works have demonstrated that direct outputs like these are not accurate measures of confidence [99]. One important area of study that can be applied to cross-view geolocalization uncertainty is particle filter uncertainty and convergence detection.

Uncertainty can be classified as either aleatoric or epistemic [41]. Aleatoric uncertainty is associated with the data that is input to the system. It is inherent to the data distribution being input, hence it is irreducible. In the context of cross-view geolocalization, aleatoric uncertainty is the uncertainty that stems from noise in our testing images. There is an intrinsic stochasticity to all measurements, even measurements that are within the distribution of our training domain. An example of an input image that might increase aleatoric uncertainty is a ground image that does not contain any unique scenery– maybe it depicts only a generic house that could be found in many locations across the search area or a type of tree that is uniformly common in the search area. Another source of aleatoric uncertainty is camera resolution– since overhead satellite image resolution does not allow the observation of the same level of detail as the ground image, knowledge is lost when converting from the real world to the sensor and hence the data does not hold enough information to allow uniquely identification of the ground-satellite image match.

Epistemic uncertainty is the uncertainty that stems from errors in the model. It occurs due to inadequate knowledge, such as a training dataset with a distribution that is not representative of the testing dataset. Epistemic uncertainty theoretically could be reduced with a better training procedure, a better model structure, or a better training dataset. An example of a source of epistemic uncertainty is training

a Siamese network on only ground and satellite images from spring, and then testing the network on images from winter. The domain shift introduced by testing on a different season than was trained on may cause higher uncertainty. Understanding and quantifying the uncertainty introduced by a domain shift like this is essential for safe use.

Aleatoric uncertainty can be approximated by characterization of the distribution of the particles that are output by the particle filter at each time step. The mean of the particle distribution can be calculated to estimate the location of a moving agent within a search area. Two options for the location estimate's use are to be reported directly to a human user or to be integrated into a larger overall navigation system. For either of these uses, it would be important to know both the estimated agent location and the uncertainty associated with that estimate. For the case of the human user, the uncertainty helps the user to understand when they can trust the location estimate. For the case of integration into a larger navigation system, the uncertainty helps to control how much influence the cross-view geolocalization estimate has on the overall localization estimate. The Kalman filter directly estimates uncertainty with a covariance matrix at each time step. However, the particle filter represents a probability distribution with its particle cloud; it does not represent uncertainty with a single number. Although particle filters do not output a numeric uncertainty estimate, the real goal of estimating its uncertainty is to determine whether the particle filter has accurately converged to the true location.

In particle filter literature, convergence typically refers to the mean squared error of the particles converging to zero– that is, the average squared difference between the particle locations and the actual agent location [19]. This definition of convergence requires knowledge of the true agent location and is mostly used in theoretical work, not practical implementations. We do not have access to the true agent location in the cross-view geolocalization application, so this definition of convergence is not particularly useful. Confusingly, the term convergence is sometimes also used to refer to the state where particles are clustered around one location, regardless of whether that is the actual agent location. When the cluster location is anything other than the

true agent location, this is often called false convergence or divergence. For clarity, this thesis uses the term "convergence" to generally refer to the particles being clustered around one location. In cases where the agent location is known, if this cluster location is the agent location then it will be referred to as true convergence, and if the cluster location is not the agent location then it will be referred to as false convergence.

Particle filters can be effective tools for approximating probability distributions by integrating sensor measurements and agent motion odometry. In some theoretical work, true convergence of a particle filter is guaranteed under a number of conditions [19]. However, in practical implementation, there must be a finite number of particles, and hence accurate filter convergence is not guaranteed. Each particle represents a potential agent state, and a finite number of particles can only represent a finite number of states. Sometimes, inaccurate motion information, bad initialization, or highly noisy measurements cause all particles to be far away from the true state. The standard particle filter then has no way to recover from this; without a restart, particles may never reach the true state [72]. Nonetheless, the resampling step of particle filtering reduces the number of unique states that are represented and hence particles tend to converge, regardless of whether they represent the true state.

Differentiating between true and false convergence is essential for reliably and accurately delivering particle filter estimates online in real-world problems, but the literature on the topic is sparse [30]. One existing metric for false convergence detection is the Kullback-Leibler (KL) divergence between the probability distributions of the current observation and of the particle cloud [97]. KL divergence, defined in Eq. 2.1, is a common measure of the difference between a probability distribution P and a reference probability distribution Q. The idea behind the KL divergence metric for false convergence detection is that in cases like the "kidnapped robot" problem and the "wakeup robot" problem, the observations will differ from the particle distribution. The "kidnapped robot" problem refers to the scenario where a robot is moved to a new location by an outside force. The kidnapped robot problem commonly causes localization system failures, as robotic systems frequently assume that the robot moves only through its own power. The "wakeup robot" problem is a spe-

cial case of the kidnapped robot problem where the robot is told that it has been moved. In these scenarios, there is a sudden change in the robot's motion and these non-smooth trajectories may not be handled well by particle filter movement models, resulting in the observations and particles having different distributions. However, this method does not account for false convergences caused by measurement noise or bad initialization, which are common sources of error in real-world particle filter deployment.

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \tag{2.1}$$

2.2 Machine Learning

Renewed focus on machine learning in the past decade has resulted in groundbreaking results on many difficult tasks. Machine learning is a technique for teaching computers tasks through examples. The computers find patterns in the example data and learn to use those patterns to inform decision making, all without explicit instruction from a human.

Computer vision is a field that focuses on making computers understand the content of visual images. Humans have adapted over thousands of years to have visual processing systems that are both sensitive to fine grain detail and capable of understanding imagery at a higher level of abstraction. Traditional, non-learning approaches to computer vision have achieved modest success, but machine learning has been responsible for the greatest recent advancements in computer vision. Artificial neural networks, commonly referred to as neural networks, are a type of model used in machine learning that aim to mimic the neurons and synapses in a human brain [43]. Neural networks typically layer multiple computational nodes or "neurons" connected by edges. The architecture of nodes and edges gives neural networks the structure that allows them to learn complicated concepts by adjusting the weights of the nodes.

There are several different approaches towards training neural networks; the training approach typically depends on the training data that is available. Supervised

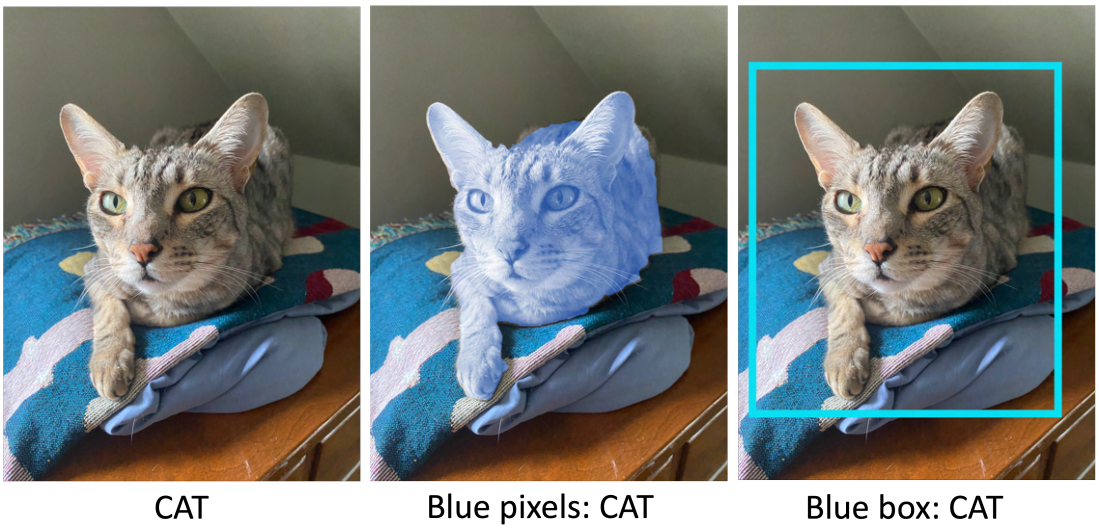
learning uses training data that is labeled with ground truth information that the network is learning to predict [49]. However, labeled data often requires human effort to provide labels and can be expensive or difficult to obtain. Unsupervised learning uses unlabeled training data that the network learns to group based on similarities, but as a result its applications are mostly limited to clustering problems [34]. Semi-supervised learning combines some labeled data and some unlabeled data to reduce labeling burden while still being applicable to different kinds of problems [118]. Supervised learning is typically used for the task of cross-view geolocalization because of the wide availability of images labeled with the locations that they were taken at.

Neural networks are trained by minimizing loss functions. The neural network takes an input from the training data and forward propagates it through the network to produce an output. In supervised learning, the loss function is used to compare the output to the ground-truth label for that input data. Backpropagation is then performed, which repeatedly adjusts the network connections to decrease the loss [83].

2.2.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of neural networks that are especially well suited for computer vision. AlexNet [50] initiated the rise in popularity of CNNs for computer vision tasks with its impressive performance at classifying images into 1000 classes based on their content. CNNs use a convolution, a type of mathematical operation, in at least one of their layers. Convolutions produce a function that describes how one input function influences another input function; in essence, they help the neural network to learn complicated relationships between visual features.

CNNs for computer vision are frequently used for either image classification or localized detection, which are illustrated in Fig. 2-2. Image classification is the task of assigning a label to an entire image based on its contents. Classification is typically used when an image contains only one relevant label. Localized detection is the task of detecting something in an image and both determining what it is and where it is within the image. Localized detection includes both semantic segmentation and object detection. Semantic segmentation is the task of labeling pixels in an image with



(a) Image classification (b) Semantic segmentation (c) Object detection

Figure 2-2: Classification and localized detection. Classification labels the entire image as containing cat. Semantic segmentation labels pixels associated with cat. Object detection detects cat and produces a bounding box of where cat is in the image.

a class. Object detection is the task of detecting a class and producing a bounding box for that class within the image.

The output layer of a classification CNN uses an activation function to directly predicts the image label. Common classification activation functions include linear, sigmoid, and softmax. The linear activation function generates an output that is directly proportional to the input. The sigmoid activation function is shown in Eq. 2.2 and is typically used for binary classification. The softmax activation function is shown in Eq. 2.3 and is typically used when there are multiple classes in an image. Sigmoid and softmax bound the output of the network between 0 and 1, so they are often treated as a probability. For example, a softmax activation function will output a vector with a value between 0 and 1 for each label class. The value for each label can be interpreted as the probability that the image contains that label.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

$$S(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_i}}, \tag{2.3}$$

2.2.2 Vision Transformers

Human visual processing systems select salient features to receive a higher level of focus in cluttered environments in a cognitive process called visual attention [98]. In machine learning, attention refers to a mechanism that helps the neural network determine what features are important, and use that relative importance of different features to improve performance. For computer vision tasks, attention typically has a spatial component. Visual attention helps the network learn the spatial layout of important features, which in turn improves the network's detection of the most discriminative features [110]. Visual attention is often incorporated as a component in CNN architectures.

Transformers take the concept of attention and build an entire network architecture around it [76]. Transformers utilize self-attention, which is a method of comparing and relating different input sequence members. Self-attention helps the network model different dependencies for each feature, building up complex relationship structures and spatial reasoning. Vision transformers (ViT) have demonstrated impressive results on computer vision tasks [24], and recent papers have studied their application to the cross-view geolocalization problem with success [115]. However, concerns remain about the level of training and computational resources required to achieve success with ViTs. This thesis focuses on cross-view geolocalization with CNN architectures, but some work that is referenced and compared against uses ViTs.

2.2.3 Siamese Networks

Deep learning-based approaches have vastly improved cross-view image matching performance [106]. Similarity learning is an area of study in machine learning that learns measures of similarity between pairs of objects, like images. Similarity learning can be applied to many different tasks, like facial recognition [32], person re-identification [17], one-shot classification [48], and image or video retrieval [108]. There are different approaches towards training similarity learning systems including classification, regression, and ranking similarity learning. In a classification approach

to similarity learning, pairs of objects are labeled as either matching or non-matching and neural network is trained to predict a binary label for an object pair. In a regression approach to similarity learning, pairs of objects are labeled with a measure of their similarity and the neural network is trained to predict a value for object pair similarity. In a ranking approach to similarity learning, the relative similarity between a set of objects is provided and the neural network is trained to predict the relative similarity between multiple objects. Relative similarity learning is well-suited towards large datasets because it requires minimal data labeling while enabling lots of data to be used in training.

The most popular approach to ranking similarity learning in cross-view geolocalization, Siamese networks, were originally designed to automatically analyze and verify written signatures [11]. Siamese networks were designed with the principle of few-shot learning, meaning that they are meant to generalize across data with only a few training examples, which is a key challenge of cross-view geolocalization. Siamese networks build off of classification networks like VGGnet [89], GoogLeNet [92], and Resnet [36] by essentially learning a new class for each ground-satellite image pair. Siamese networks consist of two identical neural networks that are trained jointly; for cross-view geolocalization, one network embeds ground images and one network embeds satellite images. A generic example of a Siamese network architecture is shown in Fig. 2-3. The jointly trained networks produce an embedding scheme that brings matching image pairs close together and non-matching image pairs far apart in embedding space. The subnetworks of Siamese networks have identical architectures and identical weights. Siamese-like networks, which have identical architectures but different weights, have been demonstrated to have slightly favorable performance over Siamese networks for cross-view geolocalization [102].

Relative similarity learning networks, including Siamese networks for cross-view geolocalization, are often trained with variations of the triplet loss. Hinge loss [15, 73] is one variation of triplet loss and is expressed as

$$\mathcal{L}_{\text{triplet}} = max(0,\ m + d_{pos} - d_{neg}) \tag{2.4}$$

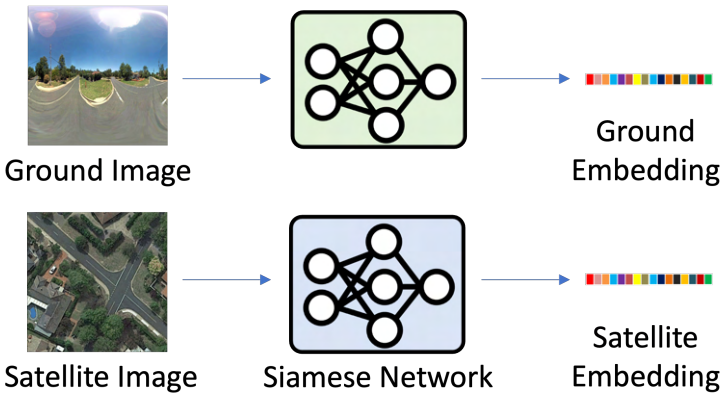


Figure 2-3: A generic Siamese network for cross-view geolocalization.

where m is a margin that determines how far apart the image pairs should be, d_{pos} is the distance between positive, matching image embedding pairs, and d_{neg} is the distance between negative, non-matching image embedding pairs. Distanced-based logistic (DBL) loss, also called soft-margin triplet loss [102] is a type of triplet loss that is more similar to softmax loss:

$$\mathcal{L}_{\text{DBL}} = \log\left(1 + e^{(d_{pos} - d_{neg})}\right). \tag{2.5}$$

The log-loss structure allows the loss to be continuous, while still reducing the loss magnitude as the distance between the positive and negative embedding pairs increases. Weighted soft-margin triplet loss,

$$\mathcal{L}_{\text{weighted}} = \log\left(1 + e^{\alpha(d_{pos} - d_{neg})}\right), \tag{2.6}$$

was proposed in [40] and shown to be beneficial by providing additional control over the strength of the loss with the α term. Binomial deviance [112] is expressed as

$$\mathcal{L}_{\text{bd}} = \log(1 + e^{-\alpha(S_{\text{p}} - m)}) + \log(1 + e^{\alpha(S_{\text{n}} - m)}), \tag{2.7}$$

where S_p is the cosine similarity between a positive image embedding pair and S_n is the cosine similarity between a negative image embedding pair. Binomial deviance enables improved training performance when there are many more negative pairs than

positive pairs, as is the case in cross-view geolocalization training. Further, [116] proposed binomial loss,

$$\mathcal{L}_{\text{bd}} = \frac{\log(1 + e^{-\alpha_p(S_\text{p} - m_p)})}{\alpha_p} + \frac{\log(1 + e^{\alpha_n(S_\text{n} - m_n)})}{\alpha_n}. \tag{2.8}$$

Binomial loss allows α and m to be independently adjusted for positive and negative pairs, hence allowing the loss function to prioritize pulling positive samples together with more strength than negative pairs are pushed apart.

2.2.4 Generalization

Neural network generalization typically refers to a neural network achieving satisfactory performance on data that was not part of its training data. However, there are many axes across which testing data may differ from training data, and neural networks may generalize well across some axes, but not across others. In [74], the authors show that neural networks generalize better on data that is more similar to the training data than on data that is less similar; i.e. generalization is not consistent across all axes. For example, in cross-view geolocalization, one axis that testing and training data might differ across could be location. A neural network may generalize from training images taken in Chicago to testing data taken in Boston. Even so, if the Chicago images and the Boston images were both acquired from Google Street View images, then they may differ in location but stay the same in other relevant axes, like weather, lighting, camera quality, season, and time of day. Compared to training images of Google Street View from Chicago, Google Street View images from Boston may be fairly similar. Many cross-view geolocalization works demonstrate generalization from one dataset to another [94, 111, 114, 119], but these are often between datasets that were made up of Google Street View images, like CVUSA [106], CVACT [56], and VIGOR [117]. Less frequently discussed is generalization across conditions that would more frequently occur between training and deployment datasets in robotics applications: weather, season, time of day, and camera.

One technique for improving generalization of image-based neural networks is

training data augmentation. Training data augmentation is a method to artificially increase the size of a training dataset for a neural network by applying transformations to the existing data. Data augmentation makes modified versions of the training data that change the appearance of an image without changing the content. Augmentations often improve neural network robustness to changes that should not affect the neural network output. Common augmentations include geometric transformations like shifting, rotating, and mirroring, color space transformations like color intensification and brightness modification, cutouts, kernel filtering like sharpening or blurring, and image mixing. Existing object [69] and place [16, 107] recognition works implement forms of data augmentation to improve generalization, but cross-view geolocalization works rarely incorporate data augmentation. Cross-view geolocalization work [105] did incorporate a variety of data augmentations and showed promising results on generalization across weather conditions, but were limited to testing on images with simulated weather changes.

2.3 Cross-view Geolocalization

Ground-to-aerial cross-view geolocalization pushes the limits of previous image matching techniques because it involves very different viewpoints. Initial works attempted to address this challenge using traditional computer vision techniques like SIFT, SURF, FREAK, and PHOW descriptors [101], which have relatively low image retrieval, semantic segmentation which with lidar in addition to images [100], image principal components and correlation, which has high computational cost [42], and finding correspondence between building facades [10], which requires operation in an urban area. Some work has combined learned or hand-tuned perception systems with robust data association algorithms [7, 8, 71], which use the relative locations of features like parking spaces to determine alignment with compelling results on small search areas.

Typical cross-view geolocalization approaches are based on the goal of one-to-one image retrieval: given a ground image and a database of satellite images, determine which satellite image is the best match. Performance is measured as the recall at top-

k, which is the fraction of ground images that successfully rank the correct matching satellite image in the top-k of the entire satellite image database. Recall at top-1 of existing approaches is rising, but is not yet 100%. One effective way to determine an accurate location estimate is to match image sequences [113], but this work still focuses on image retrieval and does not localize across long paths over time. Unlike image retrieval focused systems, this thesis focuses on maximizing the localization information gained from combining image matches over time.

Previous work [101] showed that as search area size increases, cross-view geolocalization performance degrades. Larger search areas require larger databases with a higher number of satellite images that each query ground image must be compared against. A larger database means that a cross-view geolocalization system needs to be more discriminative between matching and non-matching image pairs. This thesis uses the term "wide-area" or "city-wide" geolocalization to refer to localizing a ground image within a search area that is the size of a city or larger. Some previous works use the term "wide-area" geolocalization to mean localizing a ground image to the city that it was taken in, out of many possible cities [106], which this thesis refers to as "cross-city" for clarity. Wide-area and cross-city geolocalization have vastly different challenges; city-wide geolocalization yields many reference satellite images with very similar appearance, cross-city geolocalization yields reference satellite images with greater appearance diversity. If 100 cities are included in a cross-city geolocalization task, and all images for the correct city are ranked highly, then recall at top-1% would be high. However, this would not mean that the exact location of the ground image could be determined, just that the city could be determined. With only one city in a geolocalization task, then the Siamese network needs to learn much more discriminative and specific features to be able to tell apart one location in a city from another location in that same city.

2.3.1 Particle Filter Implementations

Several previous works have used a particle filter as temporal localization to improve cross-view geolocalization performance [40, 47, 100, 101, 109]. Using a particle filter

(a) Ground view image

(b) Matching satellite image

Figure 2-4: A centered ground-satellite image pair. Location where ground image was taken from is marked on satellite image with circle.

improves the localization performance over using image retrieval alone, but previous works that have used this approach require dense sampling of the localization area into satellite images, which generates a large database – (e.g., a satellite image for every 5 m [40] or one for each particle [47]). Those approaches must sample satellite images densely because their neural networks are designed assuming that all ground images will be centered within their matching satellite image, like the image pair example in Fig. 2-4. The large number of satellite images generated from dense sampling makes city-wide localization infeasible due to the computational and storage burden, since an embedding must be stored and a similarity must be calculated for each database image.

Most previous image retrieval-focused works [14, 54–56, 81, 87, 102] could not be combined with a particle filter that uses a coarsely sampled satellite image database, because particle filters with coarsely sampled satellite databases require matching with both centered and non-centered image pairs. Ref. [117] trained their cross-view geolocalization system to be able to recognize "semi-positive" image pairs, or ground-satellite pairs where the ground image was taken in an off-center position within the satellite image. However, their system was image retrieval focused and assumed that there may be multiple reference satellite images for each ground image. As such, they prioritized retrieving the satellite image that was closest to a centered pair with the ground image, causing less accurate image retrieval with semi-positive image pairs. A coarse satellite image-based particle filter system would only have one satellite

image for each ground image, and the Siamese network should be able to recognize a semi-positive pair just as well as a positive pair.

Although previous cross-view geolocalization works have used particle filters, little has been written about the best measurement model to be used with that particle filter. The measurement model shown in Algorithm 1, $p\left(z_t \mid \mathbf{x}_t^j\right)$, determines what weight, w_t, each particle is assigned for a given measurement, z_t. For cross-view geolocalization the measurement z_t is the Siamese network output, but there is not yet a standard measurement model that has been shown to work best for cross-view geolocalization. Existing cross-view geolocalization systems that utilize particle filters [40, 47, 100, 101, 109] either directly use image pair similarity as measurement probability as seen in Equation 2.9 or use an exponential measurement model as seen in Equation 2.10, where d_t^j is the Euclidean distance in embedding space between image pair j at time t, and B is a tuning parameter. However, Siamese network-based cross-view geolocalization systems are typically trained with loss functions that pull or push distances between positive and negative image embedding pairs like contrastive loss [47] or triplet loss [40, 56, 87]. These loss functions change embedding distance of positive pairs *relative* to negative pairs, they do not directly change the absolute distance between positive or negative pairs in isolation.

Image pair distance or similarity is not designed to stand independently as a measurement of match probability. The Siamese network training process focuses on relative distance or relative similarity between sets of image pairs. Even the metrics that are typically used to report on cross-view geolocalization accuracy reflect this; recall at top-k is a measure of how similar the correct satellite image is to the ground image relative to all of the other satellite images. Similarity or distance of an image pair gives very little information without the context of the other image pairs that are being compared against, yet existing measurement models are absolute, using d_t^j of an image pair in isolation to come up with a measurement likelihood.

$$P(z_t|x_t) = \beta d_t^j, \tag{2.9}$$

$$P(z_t|x_t) = \beta e^{-\beta d_t^j} \tag{2.10}$$

2.3.2 Narrow Field of View Ground Image Retrieval

Many previous works on cross-view geolocalization have focused on matching panoramic ground images to overhead satellite images [14, 56, 81, 87, 101, 109, 117]. Panoramic ground images are less challenging to match with because they contain more scenery and therefore more information. Overhead images contain a 360° view of the scenery around a centered ground image. Hence matching between a panoramic ground image and an overhead image is more straightforward because there is greater overlap between the scene content in both views. Additionally, panoramic cameras result in one fewer degree of freedom in matching images because orientation does not affect scene content.

Non-panoramic ground cameras make cross-view geolocalization more difficult due to the reduced number of visible features in the image and due to the matching satellite features being concentrated within one area of the satellite image instead of spread throughout it. The unknown orientation of the ground camera is a key factor in this problem. Some previous works have developed methods to incorporate an understanding of orientation into the cross-view geolocalization system, like by training the Siamese network with an auxiliary task to learn the concept of orientation and estimate it [13, 67, 102], appending orientation maps to images to be input to the Siamese network [56], by using Dynamic Similarity Matching (DSM) to calculate the correlation between ground and satellite images [88], or by jointly embedding the full satellite image as well as the satellite image portion that is visible in the limited FOV ground image [82]. Instead of jointly determining the most highly matching satellite image and the orientation, [86] assumes that the satellite image has already been determined, and they then use pose optimization to estimate the pose within that satellite image. One previous method [116] uses Grad-CAM [84] to generate activation maps of the ground and satellite images. Grad-CAM is a method to generate activation maps that indicate regions of the images that contribute most to their similarity. Since areas of high activation in the ground image are likely to correspond to areas of high activation in the satellite image, the pair of activation maps are

compared and aligned to estimate the predicted angle offset between the ground and satellite image. However, the Grad-CAM activation maps require gradient computations, which increase computational overhead. These works, not designed for mobile robotics constraints, require many search iterations, polar transformations and data augmentations at runtime and hence may be too computationally demanding for real-time robotics.

In [56], the authors assume that the orientation of the panoramic ground image relative to the satellite image will be known, and this information is encoded in the input to the Siamese network for both testing and training. They note that if previous geolocalization Siamese networks directly addressed orientation at all, they did so with only one number to represent orientation. They highlight that the geolocalization problem actually involves two spherical angles– azimuth and altitude. Altitude is essentially the pitch of a ground camera and azimuth is the yaw. To address this, they develop the idea of per-pixel orientation maps that define the azimuth and altitude for each pixel of both the ground and satellite image. These orientation maps have one channel for azimuth and one for altitude, and they are visualized as hue and saturation values respectively. The orientation maps are appended to the RGB ground and satellite images during training to provide additional supervision to the Siamese network, which improves image retrieval performance by approximately 25%. However, [56]'s work does not address non-panoramic ground images and generating these UV maps requires high levels of computation if the ground image is not centered within the satellite image.

More recent works tend to treat orientation as an aspect of the problem that can be solved at the last step [116], or do not directly encode or estimate it at all [87, 115]. In [116] orientation-invariant embedding schemes are learned through a combination of global mining, binomial loss, and training data augmentation with random rotations. Spatial Aware Feature Aggregation (SAFA) [87] is an attention mechanism that helps the network to learn image descriptors regardless of the large viewpoint difference between ground and satellite views. TransGeo [115] uses a vision transformer instead of the typical convolutional neural network (CNN) approach. This attention-focused

approach embeds patches of the images into tokens with learnable position tokens, which gives it a more flexible method for learning about orientation and position while embedding images. Although orientation-blind embedding schemes improve image retrieval when orientation is unknown, these methods do not have a mechanism by which the ground camera pose can be input when modeling with a particle filter.

The assumption that the ground and overhead images must be matched without any knowledge of the ground camera orientation makes sense if a cross-view geolocalization system is designed to be a standalone image retrieval device [13, 54, 55, 102, 106, 116]– i.e. a system where a single ground image will be input and the correct matching satellite image must be selected out of a database of possible satellite matches. An independent ground image with no context will have no associated orientation knowledge. However, the goal of this thesis is temporal localization of a moving ground agent, not standalone image retrieval. When using a particle filter for temporal localization, each particle has an associated pose.

Many of these existing works with standard ground cameras pay attention to the challenge presented by unknown orientation of the non-panoramic ground images, but their solutions do not directly address the associated challenge of position. Orientation is just one part of the difficulty of matching non-panoramic ground images. The other part of the challenge is the position of the ground agent within the satellite image. If the ground agent is centered within the satellite image then orientation gives a lot of information about where in the satellite image the ground agent will be seeing features. However, if the ground agent is not in the center of the satellite image then position also greatly impacts ground image content, as shown in Fig. 2-5.

The existing works in the field of cross-view geolocalization have achieved impressive results, and new papers with even better results are published each month. This thesis builds off of the great work that has come before it, combining key concepts from different works, to effectively apply cross-view geolocalization to autonomous mobile systems.

(a) Centered ground agent. (b) Off-centered ground agent. (c) Off-centered ground agent.

Figure 2-5: Centered and off-centered ground agents. Agent is represented by bulls-eye, orientation of camera is represented by arrow, non-visible content of satellite image is grayed out. Position greatly affects the visible content, even though all three figures have the same orientation.

2.4 Summary

This chapter has provided background on cross-view geolocalization between ground and aerial images and the associated challenges in deploying it for mobile robotics. Cross-view geolocalization is the process of matching between images of vastly different viewpoints. It is often used to perform one-off localization of a single ground image, localizing it generally to one city out of many, or localizing it precisely within a very small search area. Previous works have not localized across wide-area, city-scale search areas, which are necessary for longer term autonomy. Most prior works focus on cross-view geolocalization with panoramic ground images, whereas autonomous platforms are more likely to use limited FOV cameras. Generalization across variables other than location in prior works has also rarely been demonstrated or discussed. Additionally, previous particle filter-focused works have output a state estimate but have not provided a confidence or uncertainty measurement in that state estimate, which makes integration with a navigation system difficult.

The algorithmic solutions detailed in this thesis have been explicitly designed for localization of mobile autonomous systems. Chapter 3 introduces WAG, which trains its Siamese network with trinomial loss to improve performance on off-centered satellite images and hence enable localization with city-scale search areas. Chapter 4

introduces ReWAG, which generates pose-aware embeddings to improve localization performance on limited FOV ground images. Chapter 5 introduces ReWAG*, an extension of ReWAG that is trained with data augmentation to boost generalization performance across changes in weather, time of day, and season. Finally, Chapter 6 introduces GKL divergence tracks the particle filter distribution to measure convergence uncertainty.

Chapter 3

Wide-area Geolocalization

3.1 Introduction

Wide-area Geolocalization (WAG) is a temporal cross-view geolocalization system with advanced ability to localize across a wide, city-scale search area, enabling accurate localization in GPS-denied or GPS-degraded environments. The large search area that WAG is capable of localizing within expands the potential applications of cross-view geolocalization. WAG consists of a Siamese network and a particle filter to combine odometry with images over time to achieve accurate localization on a scale not previously demonstrated. Before deployment, WAG takes in satellite imagery of the search area and as the ground agent moves, it takes in ground imagery and odometry. Over the course of 10 to 20 km, WAG localizes down to around a few dozen meters of estimation error.

3.2 Methods

3.2.1 Overview of Approach

Wide-Area Geolocalization (WAG) [27] consists of a strategy for creating a satellite image database, a Siamese network, and a particle filter (see Fig. 3-1). This system decreases computation and storage requirements and decreases estimation error and

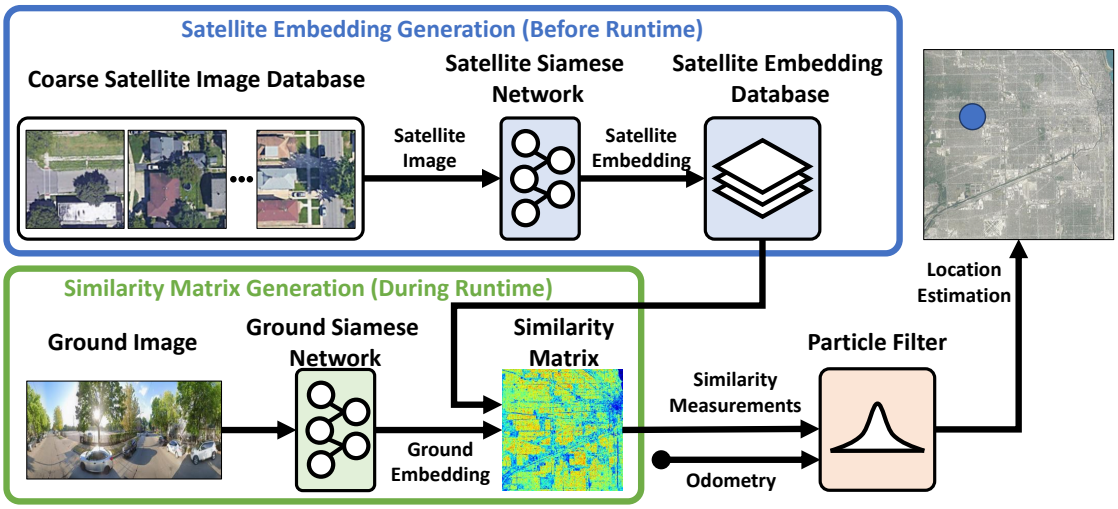


Figure 3-1: Diagram of WAG. Satellite embeddings are generated before runtime with the satellite image database and the Siamese network. During runtime, the ground embedding is generated at each time step and combined with the satellite embeddings to create a similarity matrix. Odometry and measurements from the similarity matrix are input to the particle filter with a Gaussian measurement model to generate a location estimate.

convergence time as compared to previous work [40, 47, 101]. In this chapter and in chapters 4 and 5, convergence will refer to the particle filter reaching a standard deviation and an estimation error below 60 meters. WAG requires a loosely informed guess (within approximately 2 km) of the starting state of the ground agent, a bounded search area of approximately 20 km by 20 km or less with associated satellite imagery of that area, ground-view panoramic images at each time step, and odometry between each time step.

WAG matches ground images to satellite images and uses these matches to inform its state estimate. Like many cross-view geolocalization systems, WAG employs image retrieval to perform this ground to satellite matching. This means that WAG matches image pairs, it does not match a ground image to a specific location within a satellite image. Hence, the search area needs to be discretized into a database of individual satellite images for WAG to localize. The strategy for discretizing the search area affects the size of the satellite database and the difficulty of matching between the ground and satellite images. WAG coarsely discretizes the search area to construct the database of satellite images, meaning that the satellite images it generates are not

overlapping. Section 3.2.2 describes WAG's method for satellite database generation.

WAG uses a Siamese network, described in Section 2.2.3, to match between ground and satellite images. The Siamese network is a type of neural network that generates embeddings of ground and satellite images that are similar if the images are matching. These embeddings are essentially low-dimensional vector representations of input images. Generating embeddings from the Siamese network is not instantaneous and as a result, it is desirable to generate satellite embeddings before runtime. WAG uses its Siamese network to preprocess all satellite images in the database to accelerate measurement updates. During runtime, the ground image is embedded by the Siamese network and the similarity between the ground embedding and each satellite embedding is calculated. These similarities produce a similarity matrix, which is a spatial representation of the relative similarities of the ground image with all satellite images in the search area. Section 3.2.4 describes the loss function that WAG's Siamese network is trained with to improve performance with a coarsely discretized search area.

WAG combines a Siamese network with a particle filter, a probabilistic method for recursively estimating a state as more measurements are collected. The particle filter receives measurements from the similarity matrix and odometry as input to produce a location estimate at each time step. Section 3.2.5 describes WAG's particle filter design and models.

3.2.2 Satellite Image Database Generation

Cross-view geolocalization is performed by matching from ground to satellite images. Ground images are obtained from the agent at each time step as it moves. Satellite images are obtained by delimiting a search area and discretizing that search area into individual satellite images. The search area must be discretized because WAG matches between pairs of ground and satellite images, it does not match a ground image to a location within a satellite image. The images generated from this discretization form the satellite image database, but the method of discretization affects the number of satellite images and how easily they can be matched against.

Figure 3-2: Positive pairs vs. semi-positive pairs. For the satellite image outlined in solid black, a ground image taken in the green shaded center region (ground image shown outlined in green) makes a positive pair and a ground image taken in the blue shaded region (ground image shown outlined in blue) makes a semi-positive pair. Previous works only match between positive pairs.

Previous works require dense discretization of the localization area for satellite images [40, 47] because of how their neural networks were trained. In particular, [40, 47] require specific constraints on where the ground images are taken, such as requiring ground images to be centered within their matching satellite images. Figure 3-2 shows the difference between positive (centered) image pairs and semi-positive (non-centered) pairs. In previous works, each ground image could only be matched to the positive part of a satellite image– the middle 50% of the satellite image, as shown in Fig. 3-2. Hence, the satellite database must cover the search area in overlapping

satellite images, significantly increasing the number of satellite images needed for localization within an area. However, a higher number of satellite images increases the amount of computation that must be done to match each ground image to a satellite image because at each time step the ground embedding has to be compared to the satellite images, and more satellite images requires more computation. Hence, dense search area discretization results in a system that cannot function close to real-time on large localization areas.

Instead, WAG takes an alternative approach that coarsely discretizes the localization area and enables WAG to also use the blue portion of Fig. 3-2 for localization. WAG's Siamese network is trained to match both centered (positive) image pairs and non-centered (semi-positive) image pairs, so that more of the area within a satellite image can be matched against. Thus WAG can more fully utilize each satellite image and does not need to cover the search area in overlapping satellite images. As a result of this process WAG can generate a satellite database with 10% of the images required for previous works [40].

3.2.3 Training Data

WAG is trained with the VIGOR [117] training dataset due to its inclusion of semi-positive image pairs. The VIGOR dataset consists of images obtained from the Google Maps Static API: 90,618 aerial images of size 640 $\times$ 640 taken at zoom level 20, and 105,214 panoramic ground images of size 2048 $\times$ 1024 taken at zoom level 2 of Chicago, San Francisco, Seattle, and New York City. The VIGOR dataset includes aerial images paired with positive and semi-positive ground images; positive images are taken from the center 50% of the paired satellite image and semi-positive images are taken from the outer 50% of the paired satellite image.

3.2.4 Trinomial Loss

WAG uses a Siamese network, which are described in Section 2.2.3, for matching between ground and satellite images. WAG's Siamese network consists of a neural

network for satellite images and a neural network for ground images with matching architectures. WAG's Siamese network architecture is based on that from [117], which developed the VIGOR cross-view geolocalization system. VIGOR's Siamese network uses a backbone that consists of a classic architecture for image processing, VGG-16. VGG-16 is the sixteen layer version of the convolutional neural network developed in [89] for the task of image recognition. VIGOR also incorporates the Spatial-aware Position Embedding submodule (SPE) from the spatial aware feature aggregation (SAFA) module developed in [87] to improve spatial awareness. SAFA is a module that takes feature maps from a backbone convolutional neural network, like VGG, and outputs a spatially aware feature map. A spatially aware feature map encodes spatial layout information into feature maps, which represent the output activations for different filter layers of the neural network. The feature maps are generated by combining max pooling, which selects the most prominent features, with a spatially-aware importance generator, which encodes spatial combinations of those prominent features. The spatially-aware importance generator employs self-attention, which helps it to learn which features are more prominent for the cross-view geolocalization task and their relative spatial locations within the images. Multiple spatially aware feature maps with different weights can be generated to encourage maps to focus on different features, and they can then be aggregated as one output feature descriptor. WAG uses 8 spatially aware feature maps, as VIGOR did. In [87], the authors use a polar transform on aerial images before inputting them into their Siamese network, however [117] and this thesis does not perform polar transform due to the focus on non-centered image pairs. Additionally, the polar transform adds significant computational requirements and thus is less appealing for real-time use.

WAG's neural network is trained with triplet loss (Eq. 2.5) and has additional training with a new loss function to improve its performance on non-centered semi-positive pairs and therefore enable coarser discretization of satellite images. This additional network training is performed with trinomial loss, a new loss function that builds off of binomial loss (see Section 2.2.3 for background on cross-view geolocalization loss functions) to improve matching performance with semi-positive, non-centered

image pairs. The loss function is the part of Siamese network training that measures how well the training data is being modeled by the network; lower loss means that the training data is being modeled better. The training process aims to minimize loss by adjusting the neural network parameter values. In similarity learning, the loss function determines how much to pull together positive image embedding pairs and how much to push apart negative image embedding pairs. Binomial loss [116] in Eq. 3.1 was an advancement of the binomial deviance loss from [112] that separated the tuning parameters for positive and negative pairs to increase the strength with which positive pairs are pulled together. WAG's trinomial loss takes binomial loss one step further by separating out the parameters for semi-positive pairs in Eq. 3.2's, creating:

$$\mathcal{L}_{\text{b}} = \frac{\log(1+e^{-\alpha_{\text{p}}(S_{\text{p}}-m_{\text{p}})})}{N_{\text{p}}\alpha_{\text{p}}} + \frac{\log(1+e^{\alpha_{\text{n}}(S_{\text{n}}-m_{\text{n}})})}{N_{\text{n}}\alpha_{\text{n}}} \tag{3.1}$$

$$\mathcal{L}_{\text{semi}} = \frac{\log(1+e^{-\alpha_{\text{semi}}(S_{\text{semi}}-m_{\text{semi}})})}{N_{\text{semi}}\alpha_{\text{semi}}} \tag{3.2}$$

$$\mathcal{L}_{\text{trinomial}} = \mathcal{L}_{\text{b}} + \mathcal{L}_{\text{semi}}. \tag{3.3}$$

Here the $(\cdot)_p$, $(\cdot)_n$, and $(\cdot)_{\text{semi}}$ subscripts refer, respectively, to the positive, negative, and semi-positive image pair types. Additionally, α is the weight that adjusts the how strongly each pair type is pulled together, S is the cosine similarity of a pair of embeddings, m is the margin to be used for that pair type (where margin is the goal cosine similarity for a pair type) , and N is the number of pairs of that type. $\mathcal{L}_{\text{b}}$ is the sum of a positive and negative loss term, introduced in [116]. Works previous to [117] treated semi-positive pairs as the same as negative pairs, since they assumed that each ground image would always have a perfectly centered satellite image that should be treated as its only match. This was inherent in the performance metrics used – recall at top-k measures how similar the embedding of the one true centered satellite image is to the query ground image. VIGOR [117] incorporated a new loss function to improve performance on semi-positive pairs, but our testing determined its performance drops from 41.1% recall on top-1 for positive image pairs to 3.7% recall

on top-1 for semi-positive image pairs. VIGOR incorporated semi-positive pairs in training, but it assumed that semi-positive pairs would be available in addition to positive pairs for each location, while WAG assumes that often only semi-positive pairs will be available and has more consistent performance between positive and semi-positive pairs. Our additional trinomial loss training improves performance on semi-positive pairs and enables coarse satellite image discetizing in which a ground image will not typically correspond to the center of an image in the satellite image database.

3.2.5 Particle Filter

WAG combines a Siamese network with a bootstrap particle filter to perform temporal localization. The general particle filter algorithm can be found in Chapter 2, Algorithm 1. Particle filters track many different potential states, represented by particles, and estimate the probability that each particle is the true state based on how well the particles agree with the measurements obtained at each time step. Particle filters are well suited for use with cross-view geolocalization systems because they are able to model many different distributions, even the multimodal distributions that can arise in cross-view geolocalization due to perceptual aliasing. The main components of a particle filter are propagation, the weighting, and the resampling. The details of a particle filter can vary from one implementation to another due to differences in the propagation, weighting, and resampling methods selected. This subsection will describe the models used for each of these methods.

The propagation model, also called the prediction model, motion model, or dynamics model, propagates the particles through space based off the measured motion and a motion noise model. WAG's propagation model is a basic model that incorporates noise into the baseline motion. The overall distance traveled in meters between time steps, Δ_t, is measured. Δ_t is used in conjunction with a noise percent parameter, p, and a standard Gaussian distribution $R \sim \mathcal{N}(0,\, 1)$, to calculate the amount

of noise to be added to the motion of each particle at each time step:

$$n_t^i = \Delta_t * p * r^i, \tag{3.4}$$

where n_t^i is the noise value for particle i at time step t and r^i is a value taken by the random variable R for particle i. This noise value is generated independently for the latitude and longitude directions of each particle at each time step. Each particle i at each time step t is then propagated according to:

$$\mathbf{x}_t^i = \mathbf{x}_{t-1}^i + \Delta_t * n_t^i, \tag{3.5}$$

independently in both the latitude and longitude directions.

WAG modifies the particle filter measurement model compared to prior cross-view geolocalization work to better model the noise in Siamese network outputs. The measurement model, $P(z_t|x_t)$ reflects how likely it would be to receive the current measurement, z_t from the state of a specific particle, x_t. Cross-view geolocalization Siamese networks are typically trained with triplet losses, which are calculated on a relative scale by comparing positive embedding pairs to negative embedding pairs. Previous works [40, 47] use an exponential measurement model, which are absolute. The exponential measurement model is represented as:

$$P(z_t|x_t) = \beta e^{-\beta d_t^k}, \tag{3.6}$$

where z_t is the sensor measurement for timestep t, x_t is the agent pose at t, d_t^k is the Euclidean distance between the embedding pair for satellite image k and the ground image from t, and β is a tuning parameter that is adjusted to roughly match the distribution of training data. The exponential measurement model assumes that as image pair Euclidean distance decreases, the probability of a match increases exponentially. However, a key observation of this thesis is that an exponential measurement model does not generally match the empirical data distribution for Euclidean distance between positive and semi-positive pairs. The training that is performed on Siamese

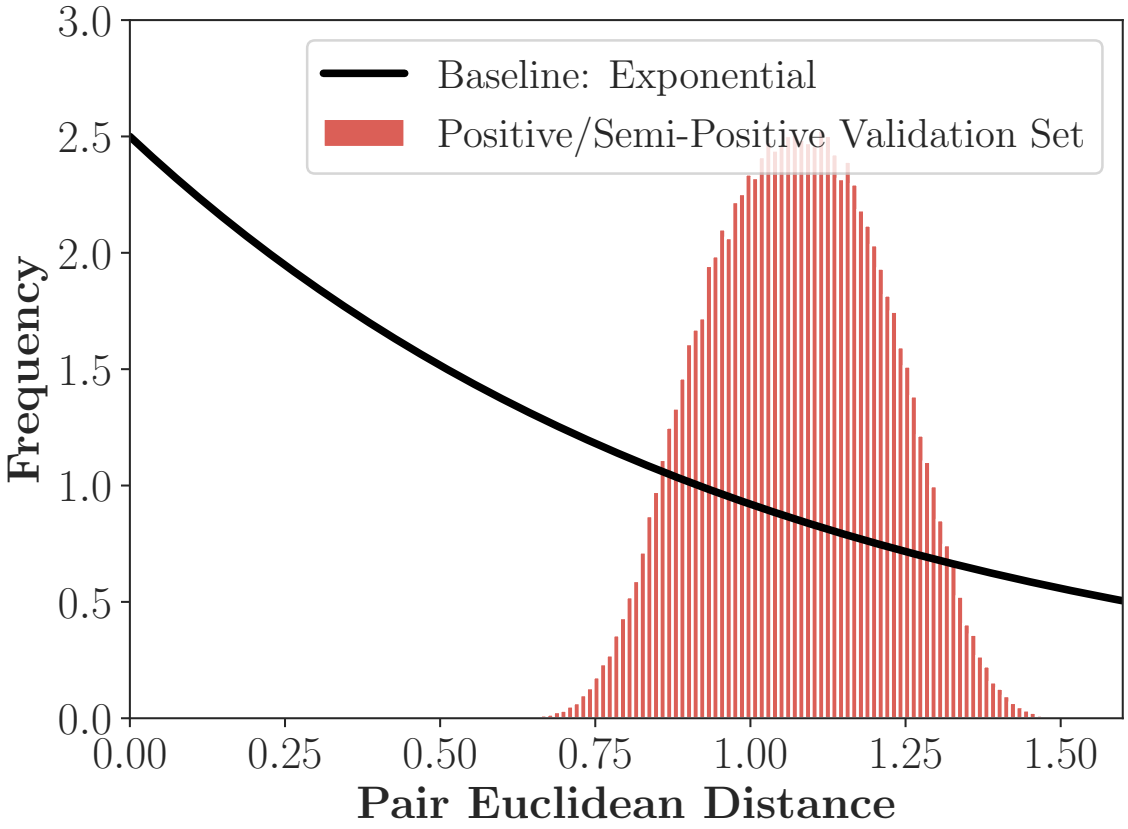


Figure 3-3: Baseline particle filter measurement model compared to empirical distribution of positive/semi-positive pair Euclidean distance measurements in the validation set from [117]'s dataset. An exponential measurement model does not fit the data.

networks with triplet loss functions is relative, not absolute, but the exponential measurement model is based on absolute Euclidean distance between an image embedding pair. As a result, the exponential measurement model does not represent the empirical data distribution well.

Fig. 3-3 shows a comparison of the actual distribution of image pair Euclidean distance in red and the exponential measurement model that is meant to represent that distribution in black. The actual distribution of the Euclidean distance between positive and semi-positive image embedding pairs for a validation set is visualized as a histogram. This section describes the method by which the empirical data was generated. The Siamese network is trained on a training set of image pairs; a validation set of image pairs is set aside so that the network does not see them during training.

After training, this validation set of image pairs is input to the network for inference and the network outputs image embeddings. The Euclidean distance between image embedding pairs is calculated. The validation set contains negative i.e. non-matching image pairs, positive i.e. matching image pairs, and semi-positive i.e. non-centered matching image pairs. Positive and semi-positive image embedding pairs should have low Euclidean distance between them, but there is a range of Euclidean distances even for matching image pairs due to noise in the Siamese network output. Previous works have modeled this noise with an exponential measurement model; this assumes that as Euclidean distance between an image pair decreases, the pair is exponentially more likely to be matching. However, the histogram of empirical data for positive and semi-positive pair Euclidean distance does not follow this exponential trend; it is normally distributed. This mismatch causes wrong estimates of the likelihood of measurements, which negatively affects particle filter convergence.

WAG treats the measurement for each ground-satellite pair as the difference between the maximum database similarity and the query pair similarity at that time step, ($z_t = \max(s_t) - s_t^j$), and uses that with a Gaussian measurement model:

$$P(z_t|x_t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\max(s_t) - s_t^j}{\sigma}\right)^2\right), \tag{3.7}$$

which better captures the output distribution of the neural network. The covariance σ was set to 0.1 to approximately fit the output distribution observed in the validation data for positive and semi-positive pairs shown in Fig. 3-4. The use of z_t as the measurement causes the distribution to be zero mean, which removes the need to find the distribution mean, another tuning parameter that would typically be required for a Gaussian noise model. Additionally, the use of a relative measurement, $z_t = \max(s_t) - s_t^j$, instead of an absolute measurement, $z_t = s_t^j$, better aligns with the loss used in the Siamese network training.

WAG uses the basic multinomial resampling method to determine which particles to replicate and which particles to remove after particles are weighted according to their similarity. Resampling algorithms use a normalized cumulative sum of weights,

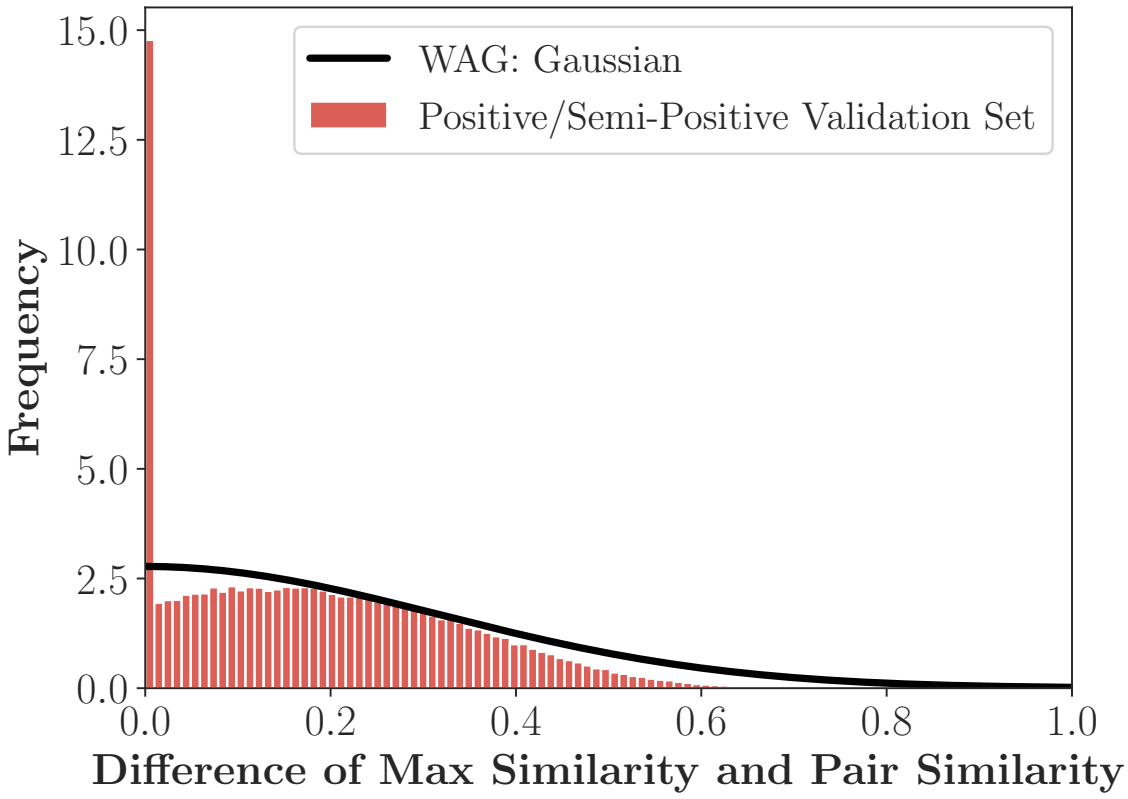


Figure 3-4: WAG's particle filter measurement model of Eq. 3.7 compared to empirical distribution in the validation set from [117]'s dataset. WAG's measurement model better fits the data. Empirical data spike at 0 is due to some pairs being the database pair with maximum similarity.

w_k, to determine which particles to keep or remove:

$$w_k = \frac{\sum_{i=1}^{k} \hat{w}_i}{\sum_{j=1}^{N} \hat{w}_j} \tag{3.8}$$

where $\hat{w}_i$ is the unnormalized weight of particle i and N is the number of particles in the particle filter. Multinomial resampling uses N uniformly distributed independent random numbers to select weights from w_k and replicate their corresponding particles.

3.3 Results

3.3.1 Experimental Setup

This thesis demonstrates localization experiments in two settings: a large scale localization with very noisy location initialization across the entire city of Chicago, and

Table 3.1: Trinomial Loss Parameters for Eq. 3.3

Parameter	**Symbol**	**Value**
Positive pair weight	α_p	5
Semi-positive pair weight	α_s	6
Negative pair weight	α_n	20
Positive pair margin	m_p	0
Semi-positive pair margin	m_s	0.3
Negative pair margin	m_n	0.7

a small-scale localization with perfect location initialization across a neighborhood in Singapore. These were experiments with simulated data: panoramic ground images and overhead satellite images from the Google Maps Static API and odometry measurements from the ground-truth displacement between images with added noise proportional to displacement. In Chicago, the satellite imagery was discretized approximately every 60 meters into a 256 $\times$ 256 grid of non-overlapping satellite image tiles at zoom level 20. In Singapore, the satellite imagery was discretized approximately every 90 meters into a 16 $\times$ 16 grid at zoom level 20. These grid sizes were chosen to maintain the same image size and resolution that the network was trained on; satellite images of size 640 $\times$ 640 pixels with a resolution of approximately 0.1 m/pixel.

In both experiments WAG has the architecture of the neural network in [117]. It is trained with [117]'s binomial loss for 30 epochs as [117] did to teach the difference between positive and negative image pairs, then trained with trinomial loss for an additional 15 epochs with the values in Table 3.1 to help it recognize semi-positive pairs. The filter in both settings has 100,000 particles and uses the Gaussian measurement model in Eq. 3.7. In the Chicago experiment, the particle filter was initialized with a Gaussian distribution centered approximately 1.3 km from the true location (standard deviation of 2.97 km) to demonstrate WAG's ability to localize with a loose initialization guess over a city-scale search area. The Chicago experiment added 2% noise to the ground-truth odometry since this experiment focuses on localizing a moving agent with poor initialization but typical odometry: 1% noise is often reliably achievable, as demonstrated in [90]. In the Singapore experiment, the particle filter

was initialized at the ground-truth location exactly to demonstrate WAG's ability to accurately track a moving agent in a small search area. The Singapore experiment added 5% odometry noise since this experiment focuses on localizing a moving agent with good initialization but higher odometry error that may cause localization drift in other systems. The noise level was higher for the Singapore experiment to make it more challenging since it was tracking an agent with a known starting location, not localizing with a very noisy initialization like in the Chicago experiment.

The following sections will detail the different experiments and results. Section 3.3.2 will demonstrate WAG's computational and storage benefits. Section 3.3.3 will demonstrate the simulated experiments comparing WAG and a baseline's localization performance on images of Chicago. Section 3.3.4 will demonstrate the simulated experiments of WAG's tracking performance on images of Singapore.

3.3.2 Computation and Storage

WAG's ability to do coarse satellite image discretization provides greater flexibility for computation and storage and enables a wider variety of applications for robotics. Since previous systems required all ground images to be centered within their satellite image pairs, for a given test area they required many satellite images. This meant many satellite image embeddings had to be computed and stored before runtime (see Fig. 3-5). The baseline for comparison used here is CVM-Net [40], which requires satellite images to be discretized densely every 5 m. CVM-Net is used as a baseline due to its impressive cross-view geolocalization performance and its demonstrated application to temporal localization with a particle filter.

The number of satellite images also affects computation because at each time step the similarity must be calculated between ground-satellite image pairs, and more satellite images means more pairs to calculate similarity between. Figure 3-6 demonstrates the timing required for a particle filter measurement update of all particles with 0.019552 ms as the estimated time for each similarity calculation (the average for our system). WAG has a similarity matrix calculation time of ≈1.28 seconds for the Chicago test area of nearly 300 km^2, while CVM-Net would take ≈3.8 minutes.

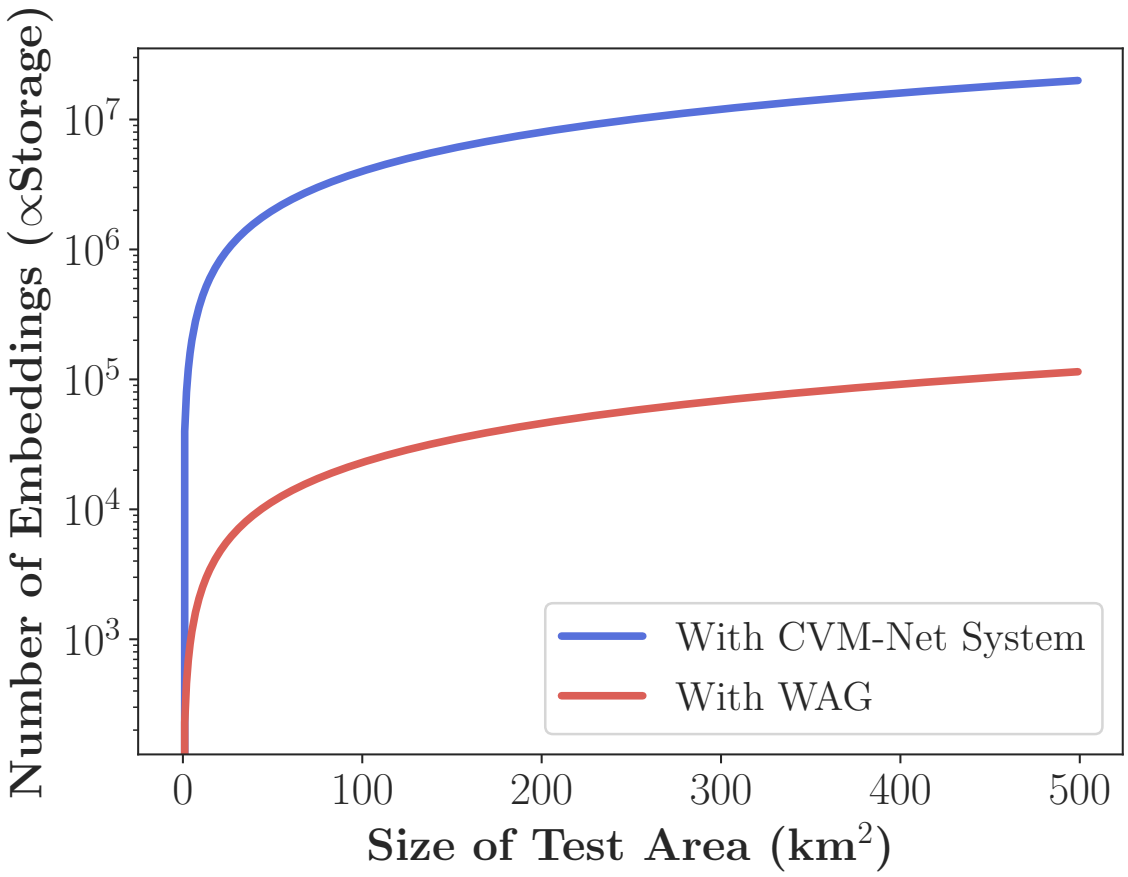


Figure 3-5: Storage requirements of WAG compared to those of CVM-Net [40]. WAG's storage requirements grow very slowly in comparison. For WAG we use a satellite image discretization density of 1 image per 66 m as used in the Chicago test, and for CVM-Net we use 1 image per 5 m as specified in [40].

WAG's timing in the Chicago test area would enable real-time localization as that test path does a measurement update approximately every 250 m, which would likely take at least 10 seconds for any ground agent to travel in a city environment. Real-time capability makes WAG useful for mobile autonomous agents to localize as they are performing other operations.

3.3.3 Large-scale Test: Chicago

Experiment details. To demonstrate WAG's ability to localize within a very large area, this thesis presents a localization experiment on the scale of the entire city of Chicago. The true simulated path is shown in Fig. 3-7, overlaid with WAG's estimated

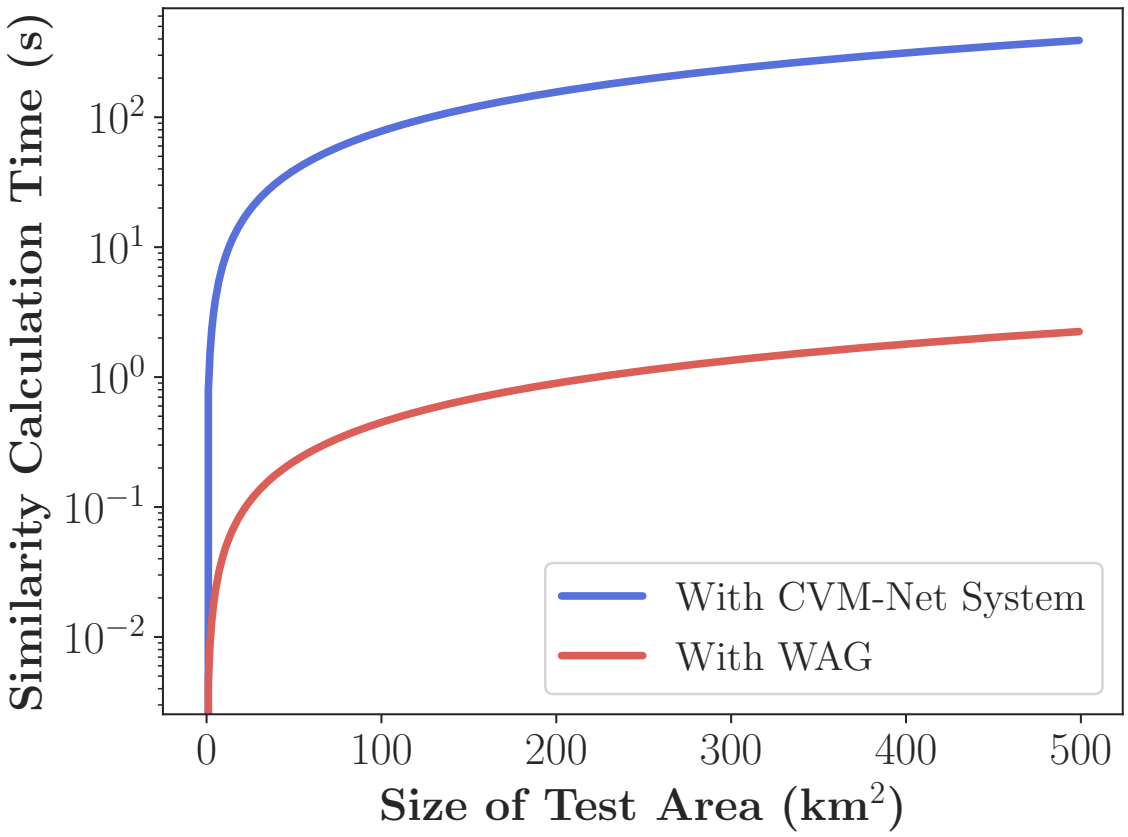


Figure 3-6: Computation requirements of WAG compared to that of CVM-Net [40] for calculation of the similarity matrix at each time step of the particle filter, with same discretization assumptions as in Fig. 3-5. WAG's computation requirements are feasible for real-time usage, CVM-Net is largely infeasible for real-time usage once a test area is larger than 10 km^2.

location at each time step. This path was generated by selecting sequential points along roads in overhead images and querying the closest Google Street View image to each point. This section will first detail the results with one path and then summarize the results with other paths that were generated in the same manner. This experiment was run with WAG and a VIGOR [117] baseline that has the same Siamese network architecture as WAG, but which is not trained with trinomial loss and which uses the measurement function in Eq. 3.6 in its particle filter (consistent with the approach in [40, 47]) instead of WAG's Gaussian measurement model in Eq. 3.7.

Estimation error. Even with an initialization 1.3 km from the agent as shown in Fig. 3-8(a), WAG has a final estimation error of 21 m, shown in Fig. 3-8(b), that

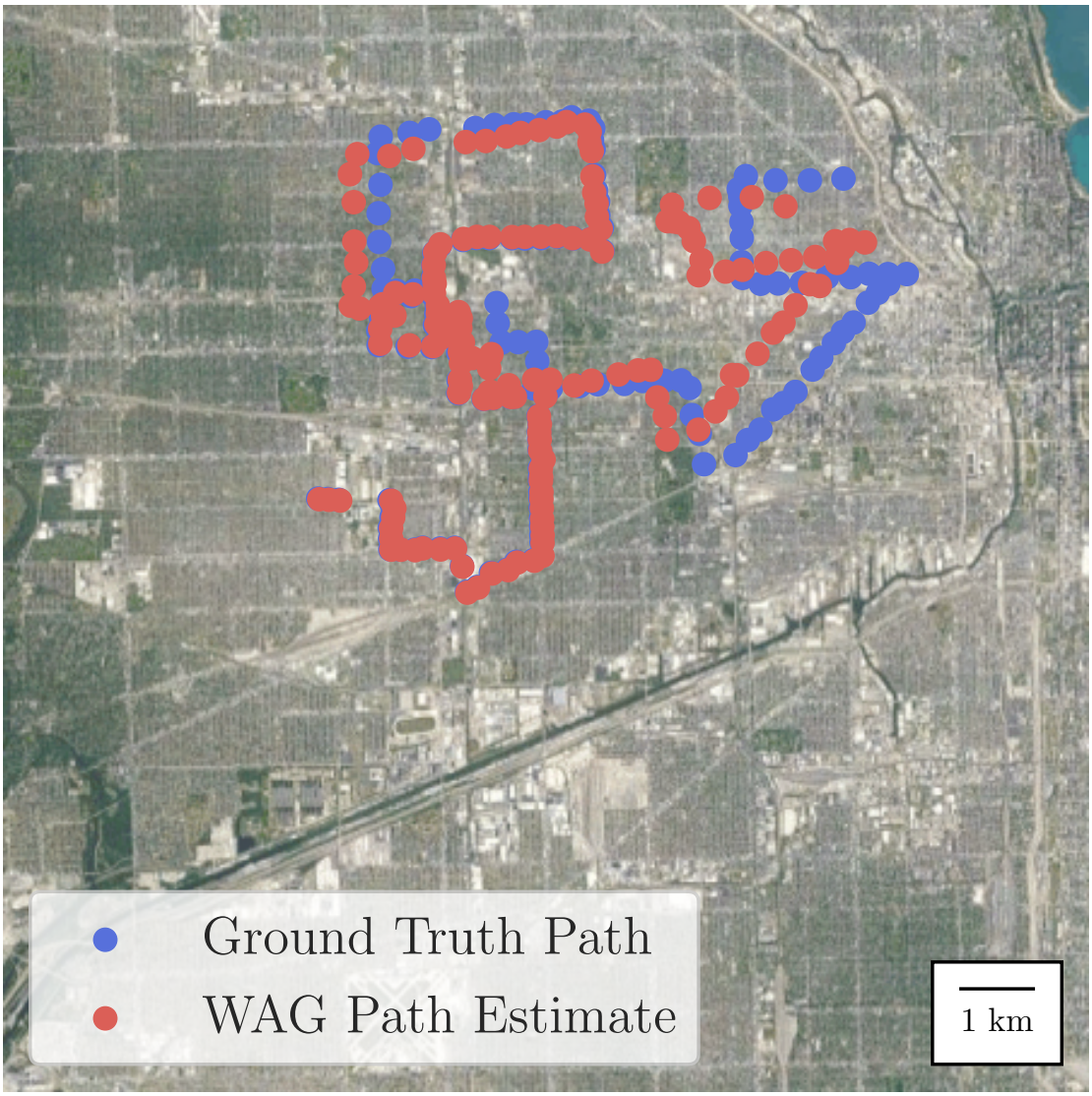
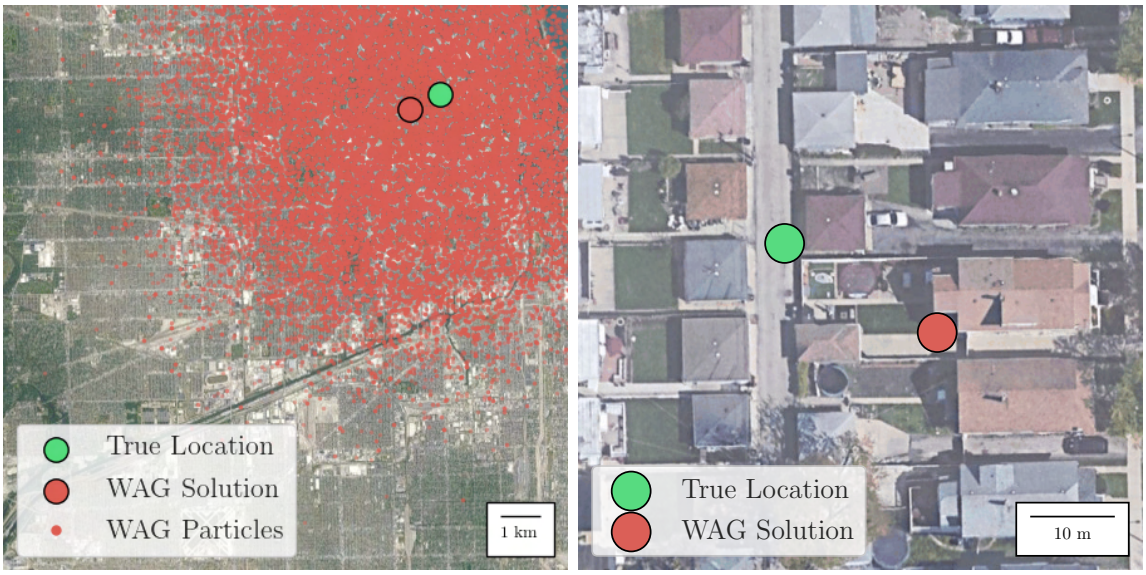


Figure 3-7: The ground-truth path in Chicago and WAG's particle filter path estimate, which accurately converges upon the ground-truth over time.



(a) Initial distribution supplied to particle filter. True location is over 1 km from initial particle filter estimate.

(b) Final particle filter solution and true location. True location is approximately 21 m from final particle filter estimate.

Figure 3-8: After it has converged, WAG is able to accurately localize an agent to within an average of 21 meters of its true location across nearly 200 km^2 of Chicago after being initialized to a Gaussian distribution centered 1.3 km from the true location.

is so small that it is comparable to the scale of the 65 m satellite tiles used for localization. Thus the estimation accuracy of WAG has essentially reached the noise floor of the system. Fig. 3-9 compares the estimation error of WAG's particle filter and the baseline as the simulated agent moves. The error is the Euclidean distance between the actual location and the localization solution of the particle filter, which is the weighted average of the locations of the particles. Over the time period shown, WAG has an average estimation error of 314 m, versus 1.5 km for the baseline. The baseline has a final estimation error of 1236 m. To the best of the author's knowledge, WAG is the only cross-view geolocalization system in the literature that localizes this accurately over such a large area.

Convergence. Figure 3-10 compares the standard deviation of the particle locations at each time step, which can be viewed as a measure of convergence [19]. In this chapter, the term convergence will generally refer to particle standard deviation. However, it is important to note that standard deviation can decrease even if estimation error does not. Chapter 6 will further discuss the nuances of convergence

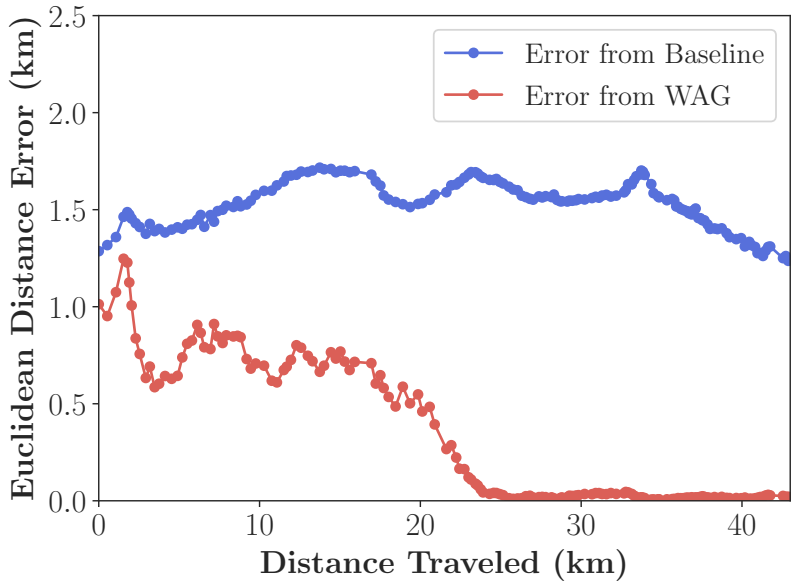


Figure 3-9: Particle filter estimation error from WAG compared to baseline system. Baseline is without trinomial loss training and with exponential measurement model. WAG has lower final and average error.

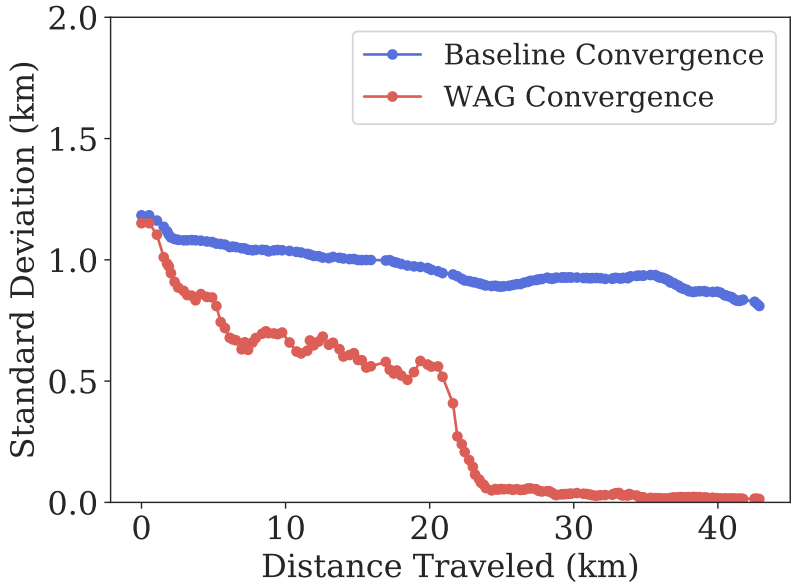


Figure 3-10: Particle filter estimation convergence from WAG compared to baseline. Baseline is without trinomial loss training and with exponential measurement model. WAG converges to an accurate estimate but the baseline does not.

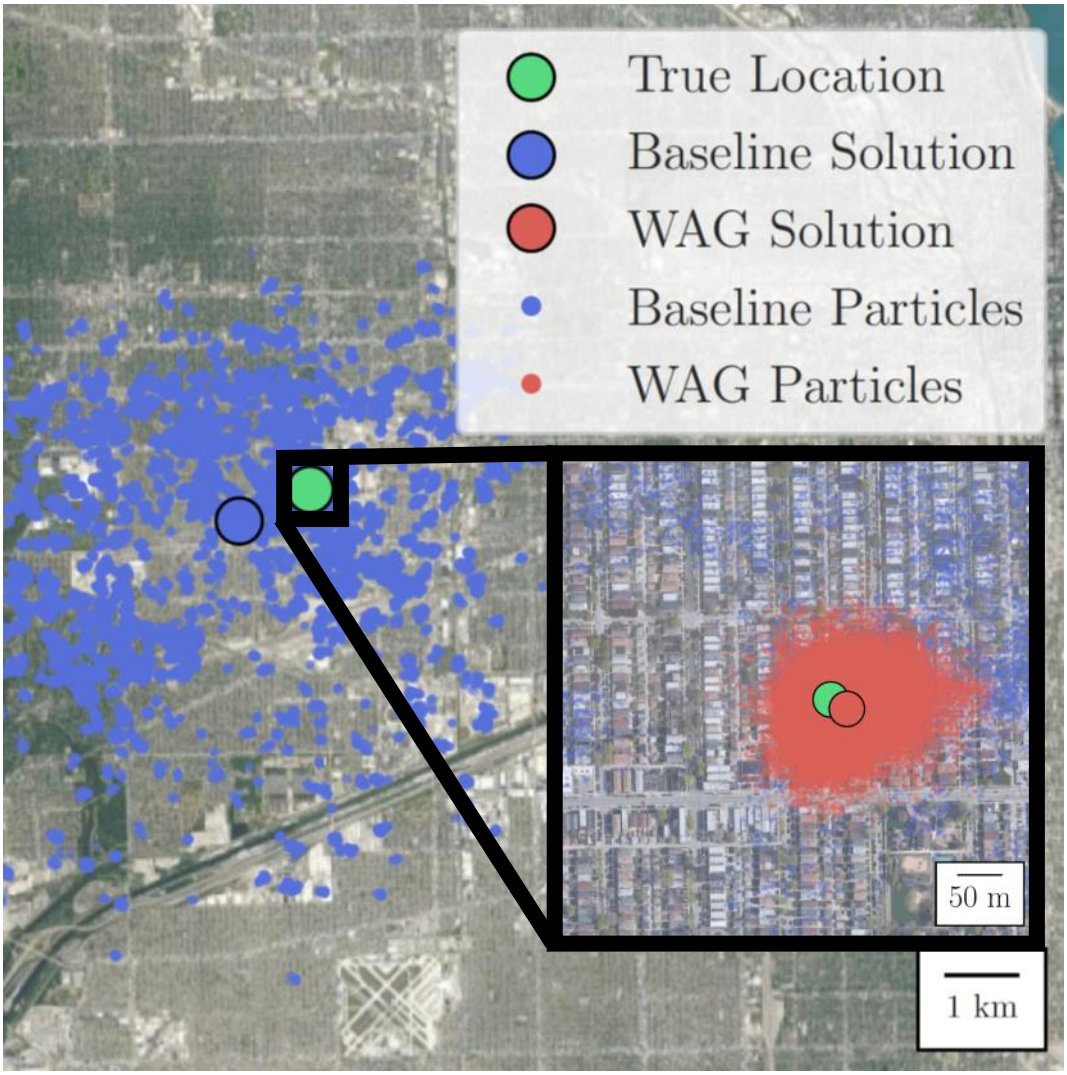


Figure 3-11: Final particle filter dispersion with the baseline system and with WAG on the Chicago test path (C-1). The baseline does not successfully converge to a location estimate, while WAG converges to within 60 meters.

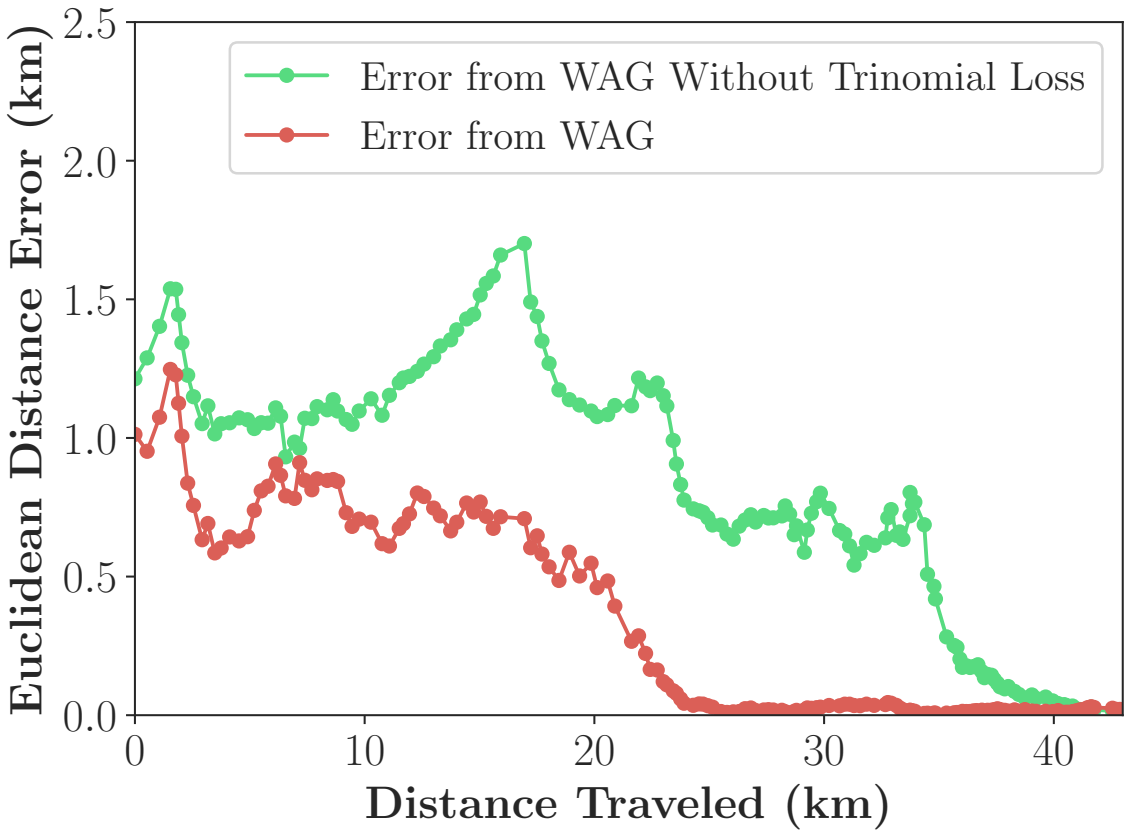


Figure 3-12: The estimation error on the same test path using [117]'s network without additional trinomial loss training.

and propose a convergence metric that is better correlated with estimation error than standard deviation. WAG converges to a standard deviation of less than 60 m (the satellite image size) after 74 filter updates, while the baseline does not reach that level of convergence in the time period tested. WAG converges to the particle filter distribution shown by the red dots in the inset of Fig. 3-11 while the baseline terminates with the particle distribution shown in blue dots in Fig. 3-11, which still shows significant estimation error.

Trinomial loss ablation. Fig. 3-12 compares the estimation error for two identical WAG systems except that one did not receive trinomial loss training. Both use the particle filter measurement model from Eq. 3.7. WAG with trinomial loss converges more quickly (74 filter updates compared to 146) and has lower average estimation error (314 m vs. 776 m). The additional training with the trinomial loss improves image retrieval performance with semi-positive image pairs and contributes towards

Table 3.2: Comparison of Results on Chicago Test Paths–
Baseline: Without trinomial loss and with exponential measurement model

Metric	System Type	C-1	C-2	C-3
Average Error (m)	Baseline	1529	1186	2290
	WAG	314	948	2130
Final Error (m)	Baseline	1236	1175	1880
	WAG	21	11	53
Convergence Time (time steps)	Baseline	-	-	-
	WAG	74	98	101

improved particle filter performance.

Multiple path result summary. Table 3.2 summarizes results comparing WAG to the baseline on three test paths in Chicago. The first, C-1, is the path discussed previously and shown in Fig. 3-7 and is a 43 km long path of 160 time steps in northern Chicago. C-2 is a 36 km long path of 104 time steps in southern Chicago and C-3 is a 41 km long path of 143 time steps in eastern Chicago. The C-2 path crossed the Chicago River once and the C-3 path crossed the Chicago River three times. Chapter 5 later reveals a particle degeneracy issue near rivers due to multinomial resampling, which likely accounts for C-3's higher estimation error. Our results demonstrate that WAG consistently outperforms the baseline in final estimation error, average estimation error, and convergence time. The baseline did not converge on any of the paths tested.

3.3.4 Small-scale Test: Singapore

To demonstrate WAG's ability to perform well on smaller scale localization tasks like those demonstrated in [40, 47, 117], this thesis simulates the experiment performed by [40], which was done in a region of Singapore called "One North". The simulated path is produced with Google Street View ground images to imitate the path from [40]; the locations of these images are shown in Fig. 3-13 with WAG's estimate overlaid. The satellite images are obtained from the Google Static API. The path is approximately 5 km long and the overhead satellite imagery that bounds it covers approximately

Table 3.3: Comparison of Results on Singapore Test Paths

Metric	System Type	Singapore
Average Error (m)	Baseline (CVM-Net [40])	16.39
	WAG	5.92

1.5 km×1.5 km. The initial location of the agent is provided exactly as in [40], so this scenario can be viewed as an odometry drift reduction test. The neural network, CVM-Net, and the particle filter implementation used in [40], were not publicly accessible so WAG's results are compared to their reported results. Ref. [40] uses visual odometry, but this simulation instead adds 5% noise to the simulated odometry, which is more than the average noise reported in [57], the source of the visual odometry algorithm used in [40]. The results in Table 3.3 show that WAG decreases the final estimation error by 64%. This simulation also demonstrates that WAG is capable of generalizing from training data of exclusively cities in the United States (Seattle, San Francisco, New York City, Chicago) to testing in Singapore, a city in a different country and continent. WAG does not need to be explicitly trained on images of the location where it will be deployed for it to accurately localize.

3.4 Summary

This chapter described how WAG performs cross-view geolocalization across city-scale search areas. WAG combines a Siamese network trained with trinomial loss and a particle filter that uses a Gaussian measurement model to localize across city-scale search areas. WAG's training with trinomial loss improves its performance matching non-centered ground images to satellite images, and its use of a Gaussian measurement model in its particle filter better matches the measurement noise, enabling fast and accurate particle filter convergence. WAG's computational and storage benefits were detailed and its performance was demonstrated on both a city-scale localization task in Chicago and a neighborhood-scale tracking task in Singapore. The biggest limitation of WAG is its reliance on panoramic imagery, which necessitates specialized hardware and limits the types of platforms that WAG can be utilized with.

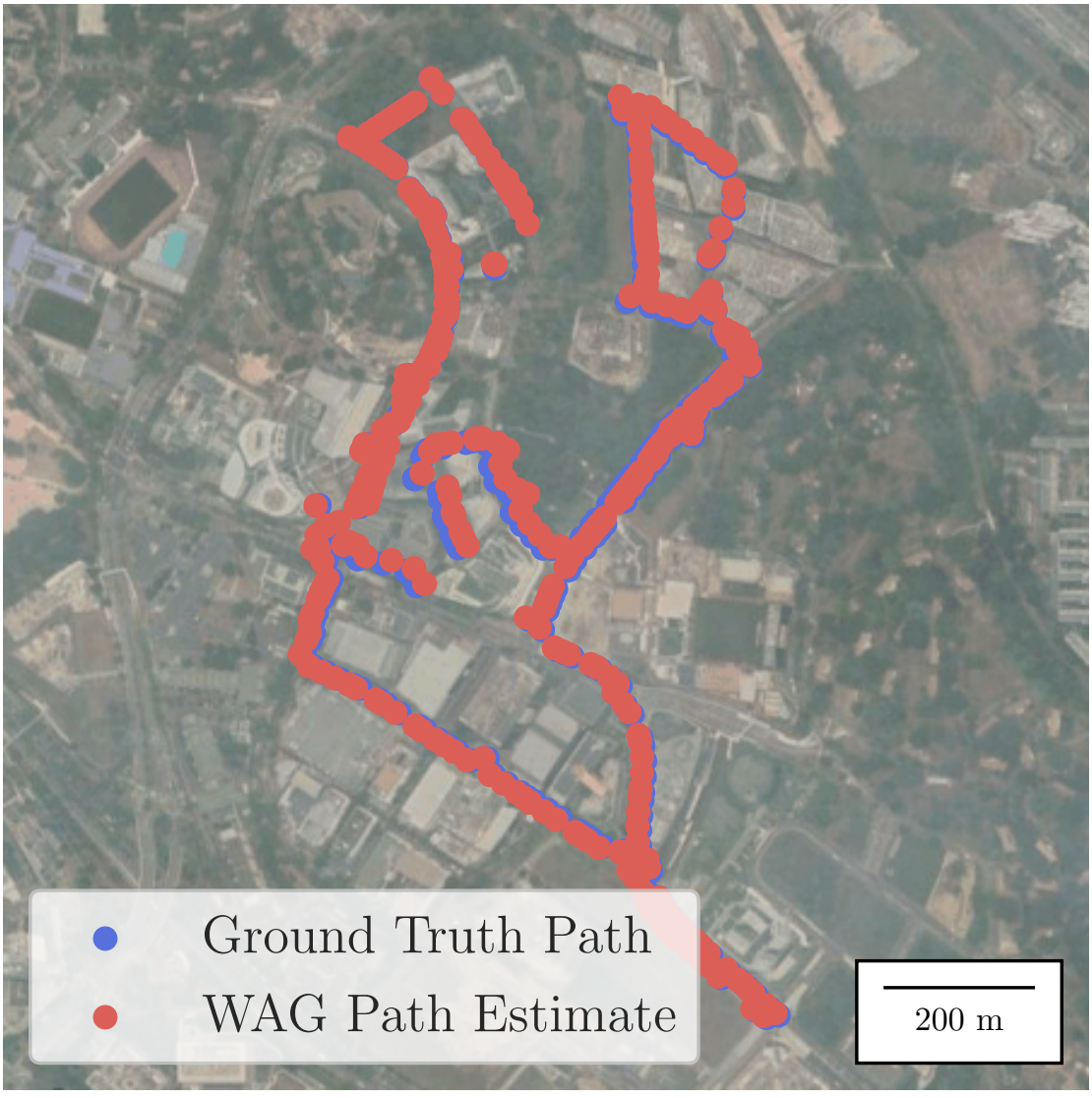


Figure 3-13: The ground-truth path in the One North area of Singapore and WAG's particle filter path estimate, which tracks the ground-truth closely.

Chapter 4 details ReWAG, which builds off of WAG to enable city-scale localization with limited FOV ground images. ReWAG generates pose-aware ground embeddings and strengthens the connection between the Siamese network and the particle filter to result in accurate localization on the same city-scale test paths as WAG.

Chapter 4

Narrow Field of View Geolocalization

4.1 Introduction

Wide-Area Geolocalization (WAG) was a big step towards increasing cross-view geolocalization scalability and improving long-term autonomy capabilities as a result, but WAG requires panoramic ground images. Panoramic ground images require either panoramic cameras, which are expensive and challenging to mount without obstructing field of view, or the stitching of multiple images, which is time consuming and computationally challenging. To improve the ease with which cross-view geolocalization technology can be adapted, it is advantageous for them to be able to operate with limited field of view (FOV) ground images.

4.2 Methods

4.2.1 Overview of Approach

Restricted FOV Wide-Area Geolocalization (ReWAG) [28] builds upon WAG [27], to enable localization across a wide search area with restricted FOV ground images, i.e. ground images with a FOV of less than 360 degrees. ReWAG uses WAG's strategies of creating a coarse satellite image database, generating embeddings with Siamese networks based on VGG-16 [89] and SAFA [87], and localizing over time with a particle

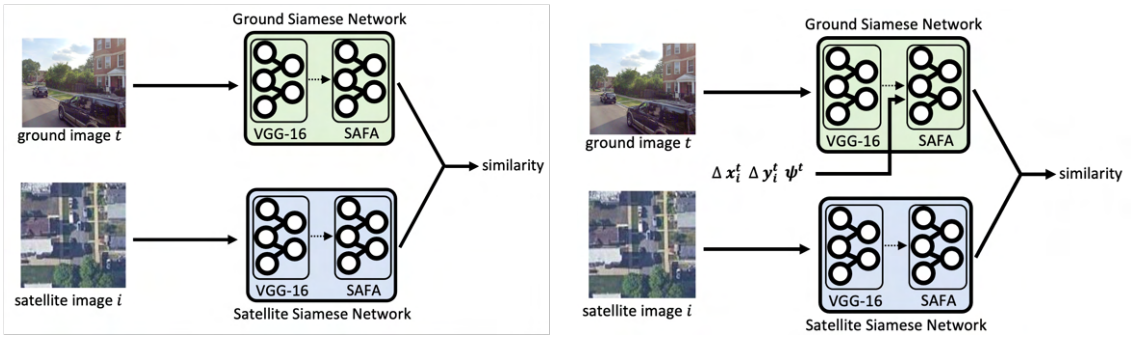
filter (see Fig. 3-1). However, ReWAG differs from WAG in two major ways. First, ReWAG generates pose-aware image embeddings in a computationally efficient manner. Second, ReWAG generates more informative similarity measures and hence more accurately models the agent location probability distribution by incorporating pose information from each particle into the Siamese network input. For non-panoramic ground imagery it is necessary to incorporate this additional information due to the increased difficulty of the problem.

ReWAG first coarsely samples the search area to construct a database of satellite images that are preprocessed with a Siamese network before runtime to generate satellite embeddings. During runtime, a generic ground embedding is generated by the VGG-16 portion of the Siamese network for each ground image, and for each particle a pose-aware embedding is produced from the base embedding and the particle's pose. The similarity between each particle's pose-aware ground embedding and its corresponding satellite embedding is calculated to produce the pseudo-similarity matrix, which is a probabilistic representation of the ground image's similarity across the search area. The particle filter receives odometry and measurements from the pseudo-similarity matrix to produce a location estimate at each time step.

Section 4.2.2 describes ReWAG's method for generating pose-aware embeddings from its Siamese network. Section 4.2.3 describes how the pose-aware embeddings are integrated with a particle filter to improve the particle filter's probabilistic representation of the state estimate. Section 4.3 demonstrate ReWAG's performance on limited FOV images in Chicago and Karlsruhe, Germany.

4.2.2 Pose-Aware Embeddings

ReWAG includes a method to train the ground Siamese network to generate pose-aware embeddings in a computationally efficient manner while modifying the network architecture slightly. Pose consists of position in x and y dimensions and orientation in heading. Like WAG, our Siamese network architecture is derived from that of [117], which consists of a VGG-16 backbone and a SAFA module to increase the network's spatial understanding. In WAG, the VGG-16 backbone outputs a three-dimensional



(a) WAG (b) ReWAG

Figure 4-1: ReWAG requires pose to be incorporated in only the SAFA portion of the ground Siamese network, which reduces computation and allows faster inference at run time.

intermediate embedding with a height, width, and depth. This intermediate output is then reshaped into a one-dimensional vector and later input to SAFA. In ReWAG, the ground Siamese network is modified to enable the generation of pose-aware embeddings. ReWAG has the same structure with VGG-16 outputting a three-dimensional intermediate embedding that is reshaped to a one-dimensional embedding. However, before inputting that intermediate embedding vector to SAFA, ReWAG appends the particle pose, as shown in Fig. 4-1. This intermediate embedding appended with the particle pose is then input to SAFA, which learns a spatial-aware representation of the ground image. The particle pose consists of three numbers: the particle's displacement from the center of the satellite image in the x coordinate frame, the displacement from the center in the y coordinate frame, and the heading of the particle. The key difference between ReWAG's architecture and previous work is that ReWAG incorporates explicit pose information into SAFA. SAFA already uses self-attention to learn spatial-aware embeddings, but ReWAG makes that spatial information more explicit by inputting pose to SAFA to create pose-aware embeddings.

ReWAG's computationally efficient benefit comes from the ability to generate one base embedding for each ground image, and then append any pose to the base embedding to efficiently generate a pose-aware embedding for each particle with the much lighter-weight SAFA. In terms of total computation, the VGG-16 backbone comprises 93% and SAFA comprises 7% of the time to produce one pose-aware em-

bedding. When combined with a particle filter, this design enables the VGG-16 inference to be done once per time step instead of once for each particle for each time step. In contrast, [82] generates pose-aware embeddings through a joint global and local pipeline, hence the full embedding generation must be done for each particle at each timestep.

4.2.3 Siamese Network and Particle Filter Integration

A central finding of this contribution is that cross-view geolocalization can be improved by more thorough integration between the Siamese network and the particle filter. Previous works have built systems with mostly one-way connections between the Siamese network and the particle filter—for each time step, the location of each particle determines which satellite image will be compared with that time step's ground image, and that similarity is used to adjust the weight of that particle. However, additional information can be integrated into the Siamese network-particle filter connection. In addition to a corresponding satellite image, each particle also has a location within that satellite image, as shown in Fig. 4-2. the particle filter is modeling a probability distribution that includes the location within the satellite image, but traditional architectures do not factor that information into the similarity measure and hence it is not reflected in the particle weights.

Particle pose information is incorporated into the similarity measure through the pose-aware embeddings. The pose of a particle i at time t consists of x and y displacements Δx_i^t and Δy_i^t, which are determined from the particle's location within its satellite tile, and the heading ψ^t, which is determined from sensor measurements at time step t. ReWAG assumes that the heading measurement will be accurate to the true ground agent heading within a few degrees based on the typical error of a compass [44]. At each time step, the ground image is used to generate one generic intermediate embedding, and for each particle at that time step a pose-aware embedding is generated from the generic embedding and each particle pose. This method increases the computation required for each particle filter update, but the computationally efficient method by which we generate pose-aware embeddings, described in

Figure 4-2: Particles are dispersed through search area, which is segmented into satellite tiles whose embeddings are precomputed before runtime. At each time step, the true heading of the ground agent, ψ^t, is approximately known and the location of each particle i within its satellite tile is given by its displacement from the center of the tile, Δx_i^t and Δy_i^t.

Section 4.2.2, helps to offset this increase. The incorporation of the particle pose aids the Siamese network in identifying where within the satellite image there should be corresponding features if the image pair is a positive match. Without pose-aware embeddings, a negative image pair would search for features to match anywhere within the satellite image, and may be more likely to match incorrectly. With pose-aware embeddings, a negative image pair will be encouraged to only look for matching features within a specified portion of the satellite image, decreasing the opportunity to find false matches.

4.3 Results

4.3.1 Experimental Setup

To demonstrate ReWAG's performance relative to WAG [27], this section performs limited FOV ground image localization experiments with the same test paths from

[27], but instead of using panoramic ground images, the ground images are cropped to 90° FOV. These test paths are a large scale localization experiment with very noisy location initialization across the entire city of Chicago using simulated data. The simulated data consists of ground images and overhead satellite images from Google Maps Static API and odometry measurements from the ground-truth displacement between images with added 2% noise proportional to displacement. The simulated data was generated in the same manner as in Chapter 3.

A stage-1 TransGeo [115] with a particle filter is used as the comparison baseline in the experiments. TransGeo uses a vision transformer [24] instead of a convolutional neural network architecture, which ReWAG and WAG use. TransGeo has demonstrated impressive performance on the VIGOR dataset (detailed in Section 3.2.3) with panoramic and non-panoramic images. TransGeo was trained in two stages, one that trains with soft-margin triplet loss (Eq. 2.5) and the second uses attention-guided non-uniform cropping to increase the resolution of important regions. The comparison only uses the first stage because the TransGeo paper demonstrated only marginal performance improvement with both stage-1 and stage-2 training, and incorporating stage-2 training would give TransGeo far more training epochs than ReWAG, which may create an unfair comparison.

ReWAG uses the neural network architecture from WAG with the VGG-16 backbone and the SAFA module, but it is modified to generate pose-aware embeddings, as described in 4.2.2. The satellite network architecture is unchanged from Chapter 3. Both ReWAG's Siamese network and the comparison baseline, a stage-1 TransGeo, are trained on the VIGOR dataset [117] with the ground images cropped to 90°. The TransGeo baseline is trained for 50 epochs with the training parameters described in [115]. ReWAG is first trained for 30 epochs with triplet loss as [117] did with α loss parameter set to 10, then trained for 15 additional epochs with trinomial loss with the parameter values described in Table 3.1. The filter has 30,000 particles and uses the Gaussian measurement model from Chapter 3. ReWAG decreases the number of particles compared to WAG's 100,000 particles because pose-aware embeddings allow the particle filter to converge with fewer particles. The particle filter is initialized

with Gaussian distributions, centered 1.3 km from the true initial location (standard deviation of 900 m) for C-1 and C-2, and centered 600 m from the true initial location (standard deviation of 300 m) for C-3. C-3 has lower initialization error because it is a more challenging path due to crossing a river several times. 2% noise is added to the ground-truth odometry and 1% noise to the ground-truth heading at each time step.

The following sections will detail the different experiments. Section 4.3.2 describes the simulated experiments comparing ReWAG and a TransGeo baseline's wide-area localization performance on limited FOV images of Chicago. Section 4.3.3 describes the simulated experiments of ReWAG's small-area localization performance on limited FOV images of Karlsruhe, Germany from the KITTI dataset [33].

4.3.2 Large-scale Test: Chicago

Experiment details. The true path the simulated agent travels in Chicago is shown in Fig. 4-3 with ReWAG's estimated location at each time step. This experiment was run with ReWAG and a baseline that uses stage-1 TransGeo [115] combined with the same particle filter as ReWAG.

Estimation error. Even with an initialization as far from the agent as Fig. 4-4(a), ReWAG has a final estimation error of 26 m (visible in Fig. 4-4(b)). This estimation error is only 5 meters greater than that which was achieved with 360° FOV ground images in WAG, demonstrated in Chapter 3. Fig. 4-5 compares the estimation error of ReWAG's particle filter and the TransGeo baseline as the simulated agent moves. The error is the Euclidean distance between the actual location and the weighted average of the particle locations. Over the duration of the experiment, ReWAG has an average estimation error of 925 m, versus 2.2 km for the baseline. ReWAG achieves a final estimation error of 26 m compared to the baseline of 2.2 km.

Convergence. Figure 4-6 compares the system convergence measured as the standard deviation of the particle locations at each time step. ReWAG converges to a standard deviation of less than 60 m (the satellite image size) after 133 filter updates, while the baseline does not reach that level of convergence in the time period tested.

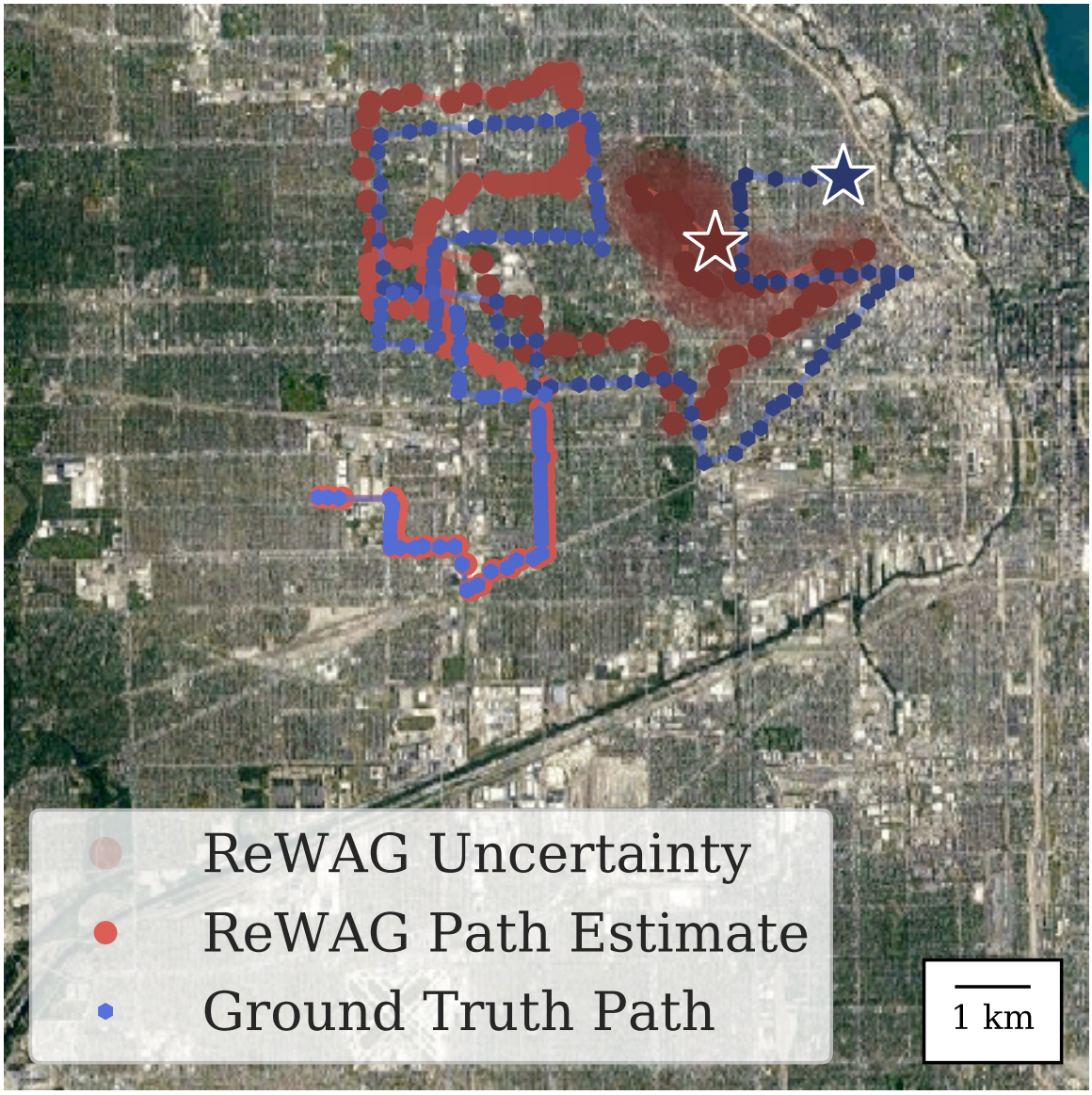
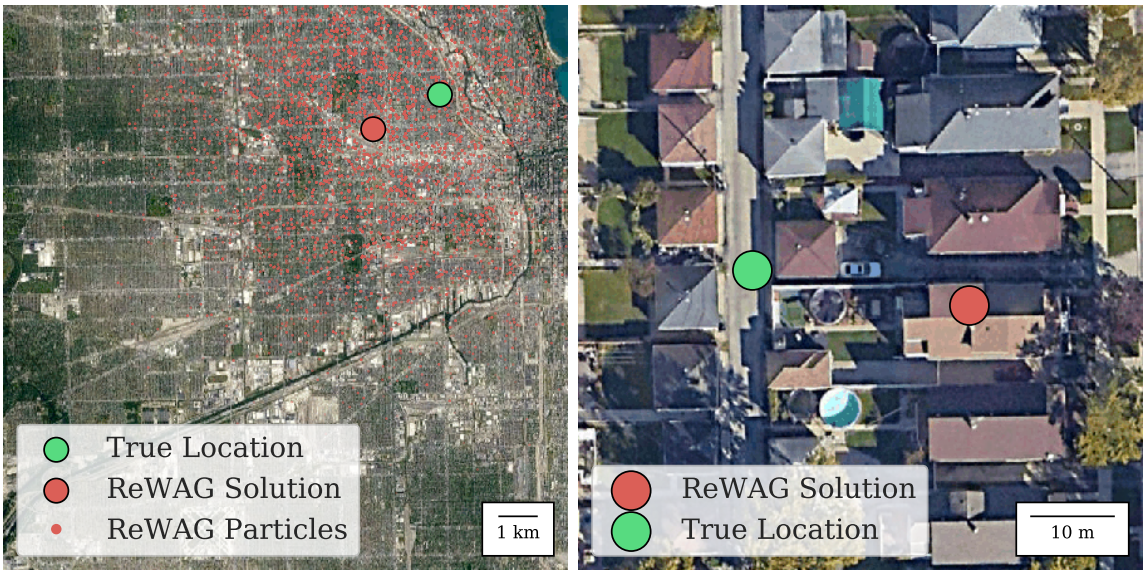


Figure 4-3: The ground-truth path in Chicago and ReWAG's path estimate, which accurately converges upon the ground-truth. Uncertainty bubble sizes are scaled down by raising to 0.75 to improve interpretability. Increasing brightness of path indicates passing of time. Start marked with stars.

Fig. 4-7 shows the particle filter distribution that ReWAG converges to as red dots in the inset of the figure while the baseline terminates with the particle distribution shown in blue dots, which still shows significant estimation error.

Ablation. This section details a small ablation study to determine the benefit of including information on both heading and the position of the particle within the satellite image in the pose-aware embeddings. Including heading and position is com-



(a) Initial distribution supplied to particle filter. True location is over 1 km from initial particle filter estimate.

(b) Final particle filter solution and true location. True location is approximately 26 m from final particle filter estimate.

Figure 4-4: ReWAG is able to accurately localize an agent to within 26 meters of its true location across nearly 200 km^2 of Chicago after being initialized to a Gaussian distribution centered 1.3 km from the true location.

pared to only including heading, or including neither as in WAG. It includes a Siamese network with the same architecture, training regime and parameters as ReWAG, with the exception that the base embeddings input to SAFA only had heading appended to them. It also includes WAG retrained on limited FOV images (without heading or position appended). The systems were tested on the C-1 test path with limited FOV images and the results are summarized in Table 4.1. The ablation shows that ReWAG improves upon WAG when used with limited FOV images. However, ReWAG with only heading, without position, performs worse than WAG. ReWAG without position's performance is attributed to the fact that heading alone does not dictate what content is visible in a ground image and hence may be misleading on its own. For example, if a ground camera is pointing north and is at the northern edge of the satellite image, as shown in Fig. 4-8(a), it is not going to contain any content that corresponds with the satellite image. If a ground camera is pointing north and is at the southern edge of the satellite image, as shown in Fig. 4-8(b), it is going to contain content that is not visible in the northern edge photo.

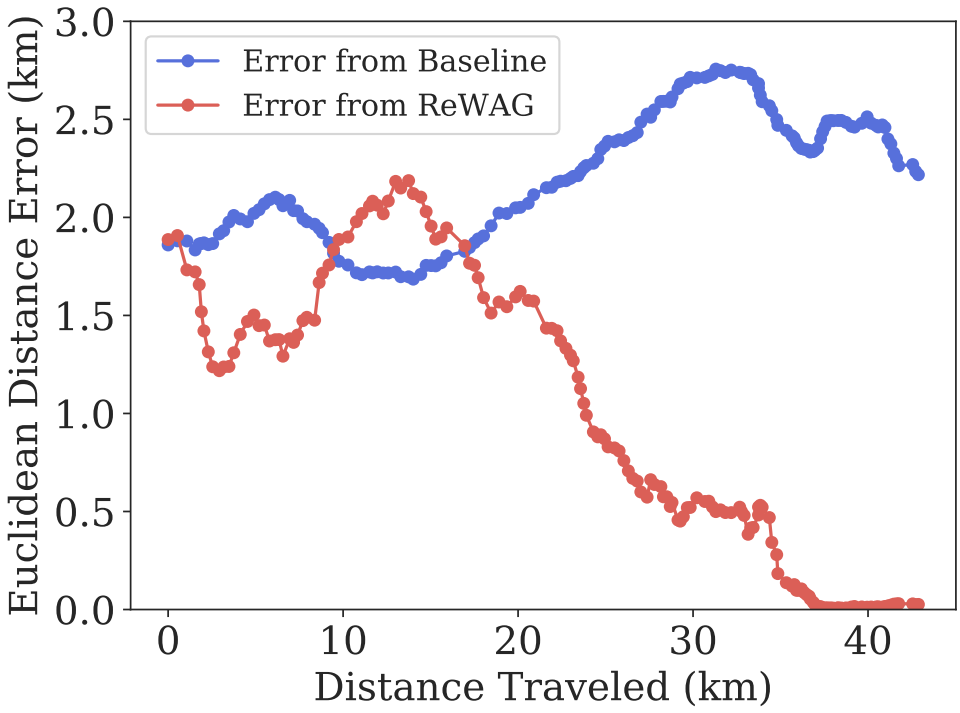


Figure 4-5: Particle filter estimation error from ReWAG compared to a TransGeo baseline. ReWAG has lower final and average error.

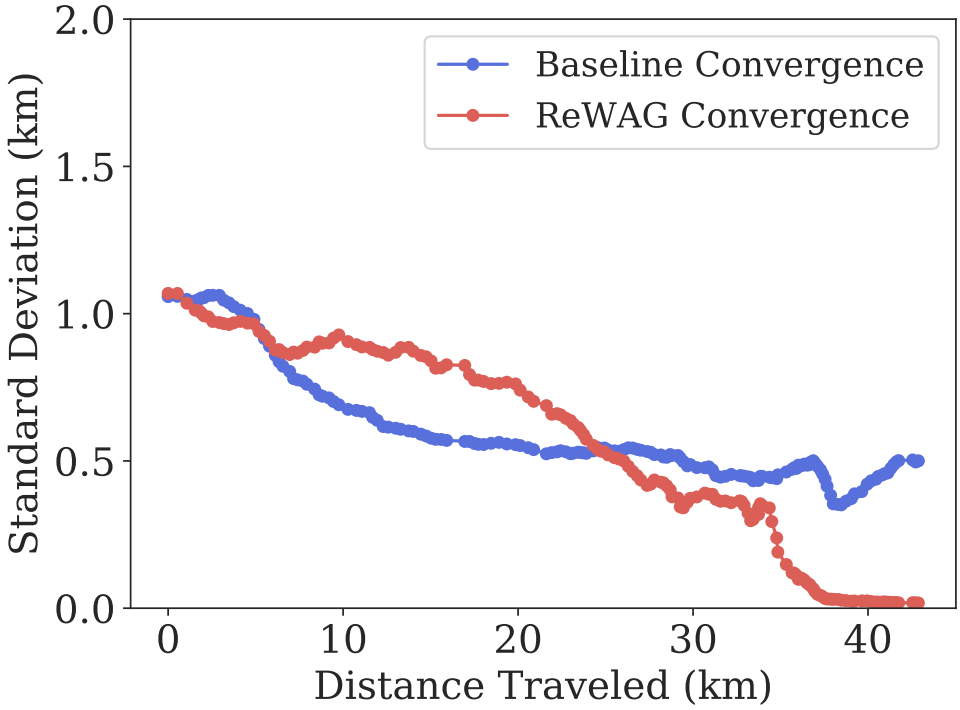


Figure 4-6: Particle filter estimation convergence from ReWAG compared to a TransGeo baseline. ReWAG accurately converges, the baseline does not.

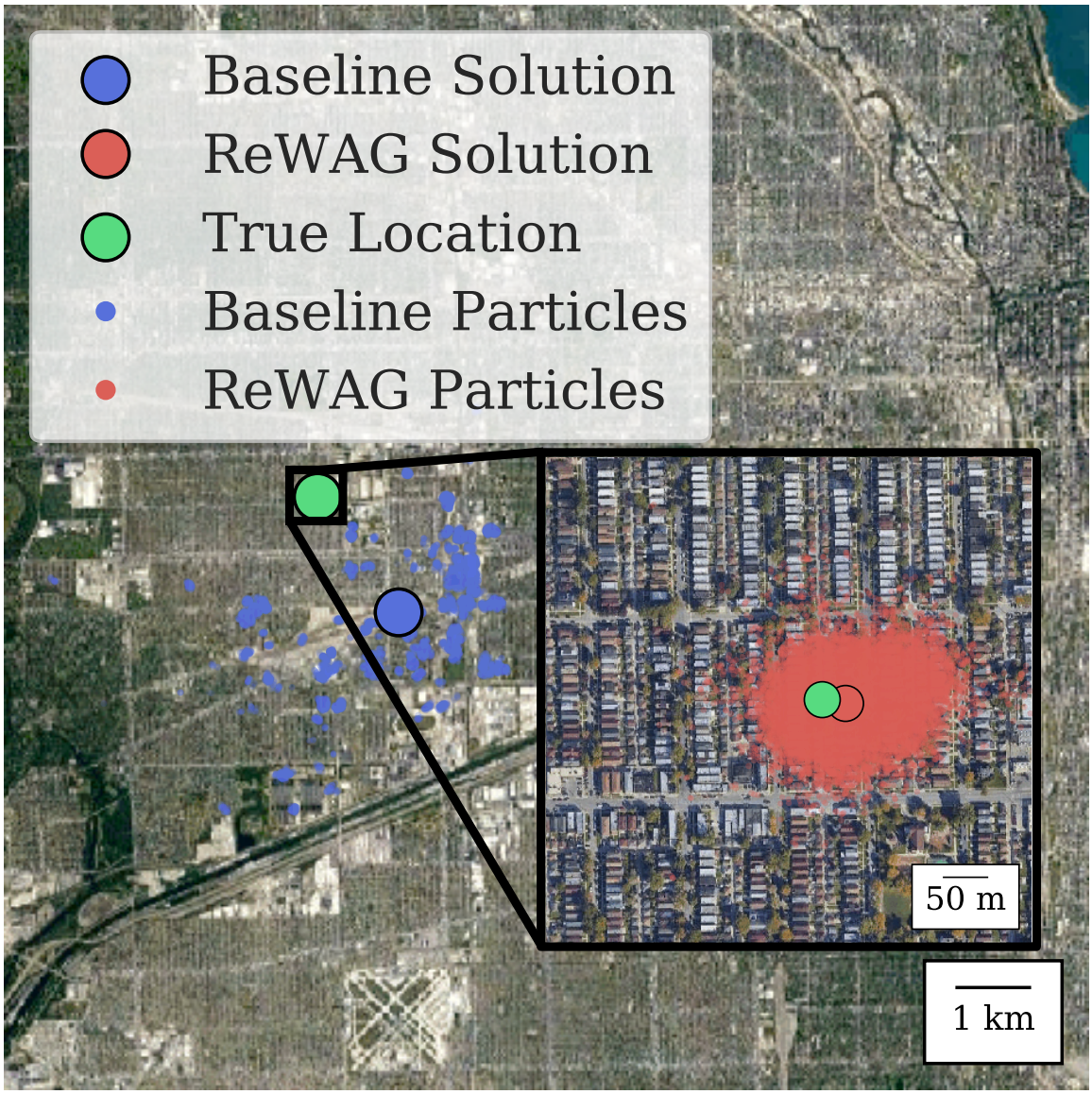


Figure 4-7: Final particle filter dispersion with the baseline system and with ReWAG on the Chicago test path (C-1). Baseline does not successfully converge to a location estimate, while ReWAG converges to within 60 m of estimation error.

Table 4.1: Ablation

Metric	System Type	C-1
Final Error (m)	ReWAG	26
	ReWAG without Position	375
	WAG	192
Convergence Time (time steps)	ReWAG	133
	ReWAG without Position	-
	WAG	-

(a) Visualization of FOV of a ground camera pointing north, located on the northern side of the satellite image

(b) Visualization of FOV of a ground camera pointing north, located on the southern side of the satellite image

Figure 4-8: The scene content that a ground camera captures is a function of both the heading and the position.

Multiple path result summary. Table 4.2 shows a summary of ReWAG's localization performance compared to the baseline on three test paths in Chicago. These paths are the same test paths that were used in Chapter 3. The first, C-1, is the path detailed previously in Fig. 4-3. The particle filter for C-1 and C-2 was initialized 1.2 km away from the ground truth, and for C-3 was initialized 600 m away from the ground truth due to the challenging path that crosses a river several times (see Chapter 5 for details on what makes a river challenging). On all paths, ReWAG outperforms the baseline in final estimation error, final standard deviation and convergence time.

Table 4.2: Comparison of Results on Chicago Test Paths–Baseline: With TransGeo

Metric	System Type	C-1	C-2	C-3
Average Error (m)	Baseline	2218	2259	300
	WAG	26	16	17
Final Error (m)	Baseline	500	1321	169
	WAG	18	10	10
Convergence Time (time steps)	Baseline	-	-	-
	WAG	133	61	41

4.3.3 Small-scale Test: KITTI

We also demonstrate ReWAG's localization performance on a KITTI test path in Fig. 4-9 [33]. ReWAG's performance is demonstrated on KITTI data because it is already widely-used and accessible for comparison and additionally it was collected in Karlsruhe, Germany, so it can demonstrate ReWAG's generalization on a city it was not trained on. This experiment demonstrates ReWAG's ability to localize with more accurate initialization information across a smaller search area and its robustness towards localizing on images in a different city than those that it was trained on. We sample images and pose data from the KITTI residential "2011_0_30_drive_0028" path and reduce the FOV to 90° by cropping the images. The KITTI dataset does not contain matching satellite images, so for the ReWAG simulation the satellite images of the search area are obtained from the Google Static API; the search area is divided into a grid of 32 × 32 satellite images with Google Static API zoom level 20. The particle filter is initialized with a Gaussian distribution approximately 80 m away from the true location to simulate localizing with low initialization error across a smaller search area. ReWAG's final estimation error is 12 m after 34 ground image updates.

This simulation demonstrates ReWAG's geographic generalization ability. Similar to WAG, ReWAG was trained on VIGOR's [117] training dataset consisting of images from Seattle, San Francisco, New York City, and Chicago. ReWAG is able to successfully localize on images of Karlsruhe, Germany, with training on images exclusively of the United States. This simulation also demonstrates ReWAG's generalization to

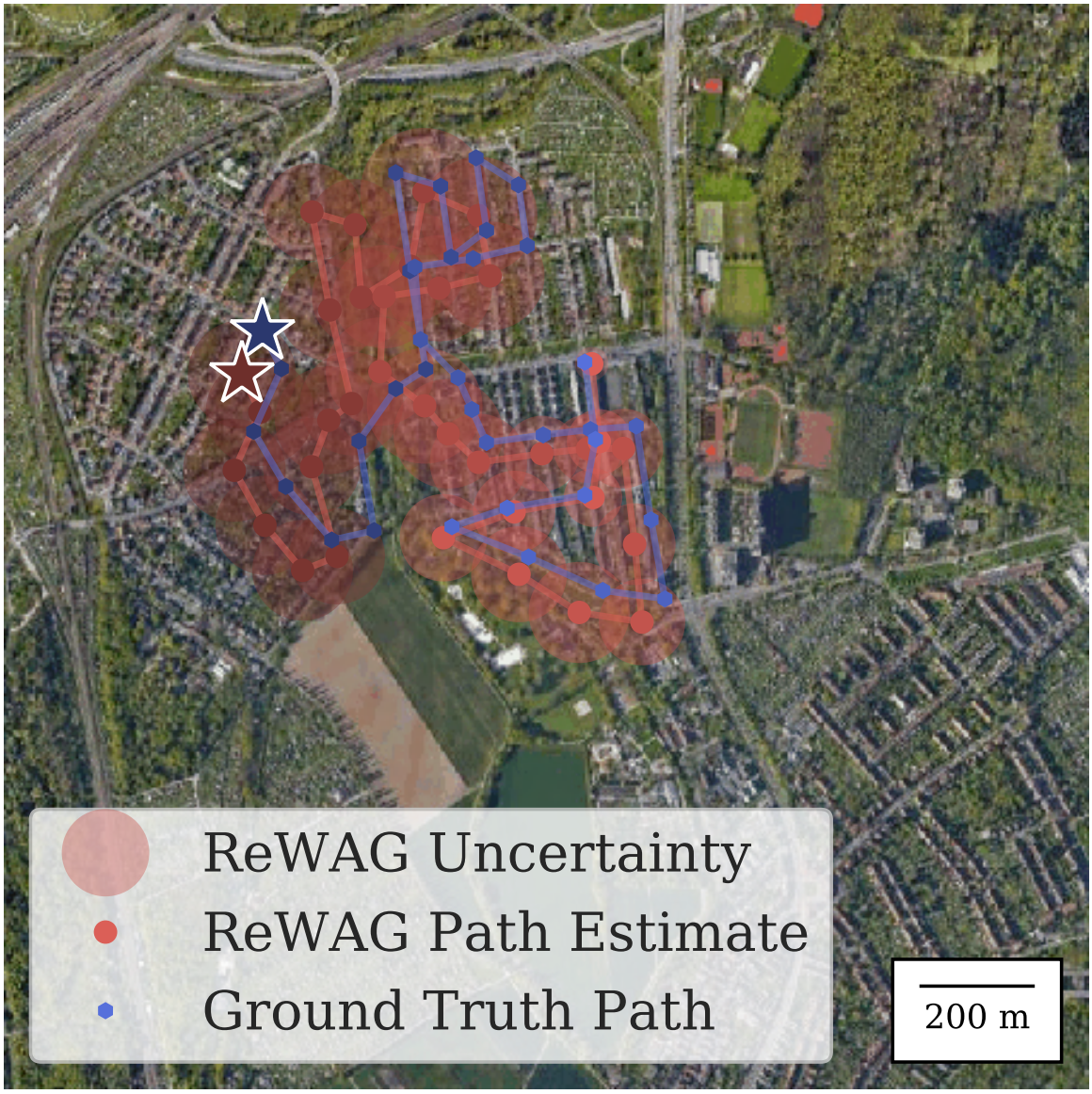


Figure 4-9: Ground-truth and ReWAG's estimated path in the KITTI test area. Increasing brightness of path indicates passing of time. Start marked with stars.

images taken with a different camera: the training images from VIGOR are cropped panoramas from Google Street View, while the testing images were taken with a Point Grey Flea 2 camera.

4.4 Summary

This chapter described how ReWAG performed wide-area cross-view geolocalization on limited FOV ground images. ReWAG incorporates pose into its Siamese network to generate pose-aware embeddings, and uses those pose-aware embeddings in conjunction with a particle filter to improve localization performance on 90° FOV ground images. ReWAG's pose-aware embeddings also only incorporate pose into the last few layers of the Siamese network, which requires less computation than incorporating pose throughout the entire Siamese network. ReWAG demonstrated faster and more accurate localization than a baseline on limited FOV images on a city-scale localization task in Chicago and a neighborhood-scale localization task in Karlsruhe.

ReWAG expands the platforms that wide-area cross-view geolocalization can be applied to by enabling the use of non-panoramic ground cameras, but it leaves room for improvement in generalization to realistic environments. Chapter 5 details ReWAG*, an extension to ReWAG that improves its generalization performance to different weather, seasons, and time of day than training data. ReWAG* includes additional training data augmentation and modifications to particle filter strategy beyond ReWAG. ReWAG*'s performance is demonstrated on a realistic, challenging testing dataset collected in the Boston area across different seasons than the training data or the satellite imagery.

Chapter 5

Experimental Hardware Demonstration

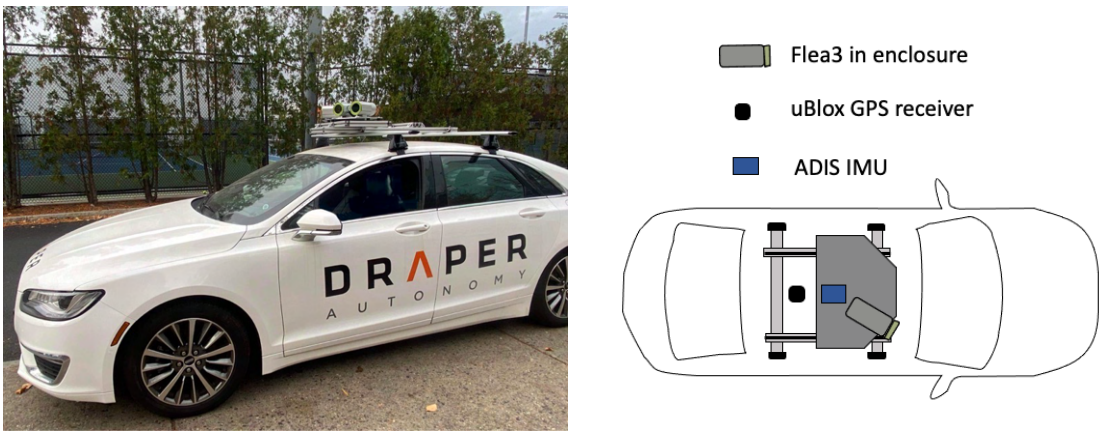
5.1 Introduction

WAG and ReWAG both contributed impressive advances towards cross-view geolocalization: WAG with its ability to perform geolocalization across a city-scale search area and ReWAG with the additional ability to localize with a limited field of view camera. However, both WAG and ReWAG struggled with some aspects of realistic simulations. The Chicago simulations were fairly similar to the data that WAG and ReWAG were trained with, hence did not push WAG or ReWAG on generalization. This chapter will further explore some of the weaknesses of ReWAG and propose ReWAG*, an improved iteration of ReWAG, to resolve those issues.

5.2 Methods

5.2.1 Overview of Approach

ReWAG* [29] extends ReWAG, improving its localization performance on realistic data to enable localization generalization across different conditions. ReWAG* improves image matching performance in situations of variable lighting by introducing



(a) Photo of car with sensors.

(b) System diagram.

Figure 5-1: Experimental setup of platform used for data collection.

data augmentation in the Siamese network training. ReWAG* also reduces the effects of particle degeneracy that arise when testing paths alongside bodies of water by implementing systematic particle filter resampling instead of the multinomial resampling that ReWAG uses in Chapter 4.

Section 5.2.3 describes the training data augmentation that is performed with Fancy PCA to produce images with different brightness. Section 5.2.4 describes improvements to particle resampling to improve robustness.

5.2.2 Dataset Collection

This thesis includes a dataset of approximately 6 hours of driving data in Boston, and Cambridge, Massachusetts to further test ReWAG; the data collected is available online at https://doi.org/10.5281/zenodo.7818704. This Boston data initiated changes that resulted in the evolved ReWAG, ReWAG*. It was collected driving a Lincoln MKZ (see Fig. 5-1) equipped with a Point Grey Flea3 camera with Navitar 4.5 mm effective focal length lens, an ADIS 16448 IMU, and a uBlox GPS receiver. The camera was mounted within a weatherproof enclosure on a plate on top of the car, angled approximately 30° to the passenger side. The IMU was mounted on the underside of the plate within an EMI shielded enclosure, centered on the car. The GPS receiver was mounted on the roof of the car behind the camera plate. Our dataset consists of

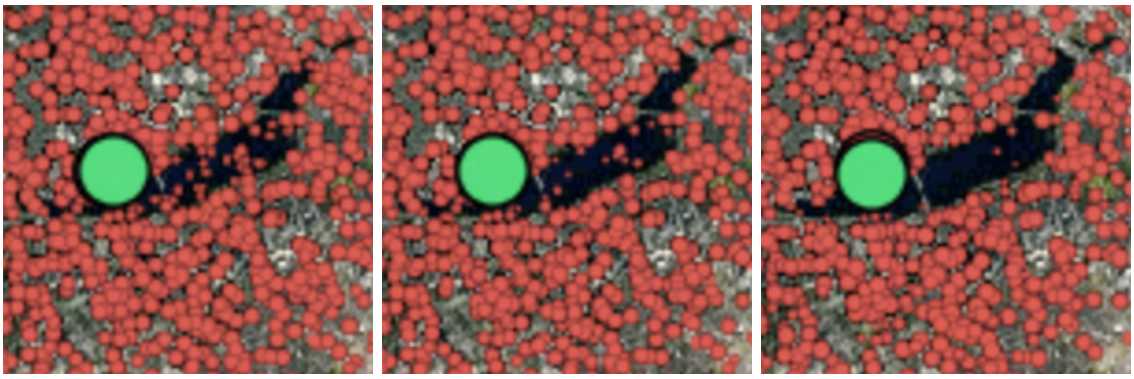
(a) Google Street View image of Amherst Alley, Cambridge, MA.

(b) Collected image of Amherst Alley, Cambridge, MA on a cloudy day.

Figure 5-2: Example of difference in appearance between Google Street View image and self-collected image. Google Street View images are carefully selected and processed, while in practice weather conditions may vary, and camera exposure and aperture may not be perfectly adjusted.

rectified RGB camera imagery taken approximately every 5 seconds along the driving paths and the corresponding GPS tags for those images. We obtained the GPS tags using SAMWISE [90], which fuses the GPS data from our uBlox receiver with visual-inertial odometry to obtain smoothed estimates of the precise location where each image was taken, even in urban canyons and tunnels with poor GPS estimates. We chose to angle our camera 30° to the right side of the vehicle because it shows more of the surrounding scenery instead of the center of the road, which is less distinctive and may often be partially blocked with other cars. Satellite images were collected from Google Maps in the same manner as in the Chicago simulation in Chapter 3; the satellite imagery was discretized approximately every 60 meters into a 256×256 grid of non-overlapping satellite image tiles at zoom level 20. The Boston satellite images were captured in July 2022.

The first tests on the Boston data revealed two key weaknesses of our initial version of ReWAG. First, ReWAG was sensitive to the brightness of the ground images it was tested on. The B-1a and B-1b test paths, which were collected on an overcast, cloudy day, highlighted this sensitivity. Originally, the Siamese network was trained on satellite images from Google Maps and ground images from Google Street View without any data augmentation. However, Google Street View images are not



(a) Time t. (b) Time t + 1. (c) Time t + 2.

Figure 5-3: Example of particle degeneration in Charles River. Red dots represent particles, green circle represents true location.

randomly dispersed in different weather conditions, they are overwhelmingly taken on bright and sunny, or partially sunny days, (see Fig. 5-2). Previously published Siamese networks for cross-view geolocalization, including WAG (Chapter 3) and ReWAG (Chapter 4), had not done data augmentation during training. This lack of training on darker or lower exposure images made cloudy ground images out of the training distribution for ReWAG's Siamese network, and because ReWAG was not familiar with these kinds of images, it could not discriminate between matching ground-satellite pairs and non-matching pairs. This prompted the incorporation of data augmentation into ReWAG*'s training.

Secondly, ReWAG suffered from particle degeneracy that became particularly apparent in the B-1b and B-3b test paths, which spent significant time traveling on roads directly adjacent to the Charles River. Particles that were propagated into the river quickly degenerated and were never recovered, as illustrated in Fig. 5-3. This made it unlikely for a path along the river to be accurately estimated, since noisy odometry makes it likely for particles on the roads adjacent to rivers to eventually be propagated into the river at some point in their trajectories, causing them to then degenerate. The testing on these challenging paths prompted improvements in the particle filter resampling strategies for ReWAG*.

5.2.3 Training Data Augmentation

ReWAG* is trained with data augmentation to improve its robustness to image brightness changes relative to ReWAG [28]. Cross-view geolocalization is sensitive to spatial alignment; for example, a satellite image that's been cropped to only include its northwest region may lose the part of the image where its ground image pair was taken, breaking the match. Due to this spatial alignment sensitivity, most previous works have not incorporated data augmentation into the Siamese network training pipeline. Although spatial transform augmentations are not particularly helpful for cross-view geolocalization, brightness augmentations are. A Siamese network should be able to identify the matching satellite image for a given ground image no matter how dark or bright that ground image is, as long as the scene is still discernible.

One method for brightness augmentation that results in brightness variation that has natural appearance is Fancy Principal Component Analysis (Fancy PCA). Fancy PCA is a method for altering the intensities of pixel values in an image, done by adding random multiples of the image principal components to each image channel. The use of image principal components to control brightness is what allows Fancy PCA to generate more images with more natural brightness variations. Each RGB image pixel is represented by $I_{xy} = [I^R_{xy}, I^G_{xy}, I^B_{xy}]^T$, where I^R_{xy} is the image pixel value at location x,y for the red channel, I^G_{xy} is the value for the green channel, and I^B_{xy} is the value for the blue channel. Fancy PCA is performed to generate augmented pixel values, I'_{xy}, as such

$$I'_{xy} = [I^R_{xy}, I^G_{xy}, I^B_{xy}]^T + [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]\,[\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T \tag{5.1}$$

where p_i and λ_i are the ith eigenvector and eigenvalue of the 3×3 covariance matrix of pixel values and α_i is a random variable that controls magnitude. The variable α_i is drawn from a Gaussian distribution for each image. The PCA components are added to every pixel of each image. Fancy PCA produces brightness changes with natural appearances, as some examples in Fig. 5-4 show. ReWAG* is trained with Fancy PCA for the 15 epochs that it is trained with trinomial loss to allow the Siamese

(a) Original image.

(b) Image modified to off-nominally increase brightness with Fancy PCA.

(c) Image modified to nominally increase brightness with Fancy PCA.

(d) Image modified to off-nominally decrease brightness with Fancy PCA.

Figure 5-4: Examples of original image and images modified with Fancy PCA brightness augmentations at both nominal and off-nominal levels.

network to learn image matching on easy, consistent brightness pairs before learning to match harder pairs with inconsistent brightness.

5.2.4 Particle Filter Resampling

ReWAG* also benefits from improvements to its particle filter resampling strategy relative to Chapter 4. Particle filter resampling is done after the reweighting step of the particle filter in Chapter 2, Algorithm 1. Reweighting converts the weighted distribution of particles to an unweighted distribution by retaining and replicating particles with high weights, removing particles with low weights, and equalizing the particle weights.

Different strategies for resampling determine the method by which particles are selected for replication or removal. The most straightforward way to resample is multinomial resampling, as was done in [27, 28]. This approach is relatively simple to implement, but it does a poor job of selecting particles uniformly relative to their

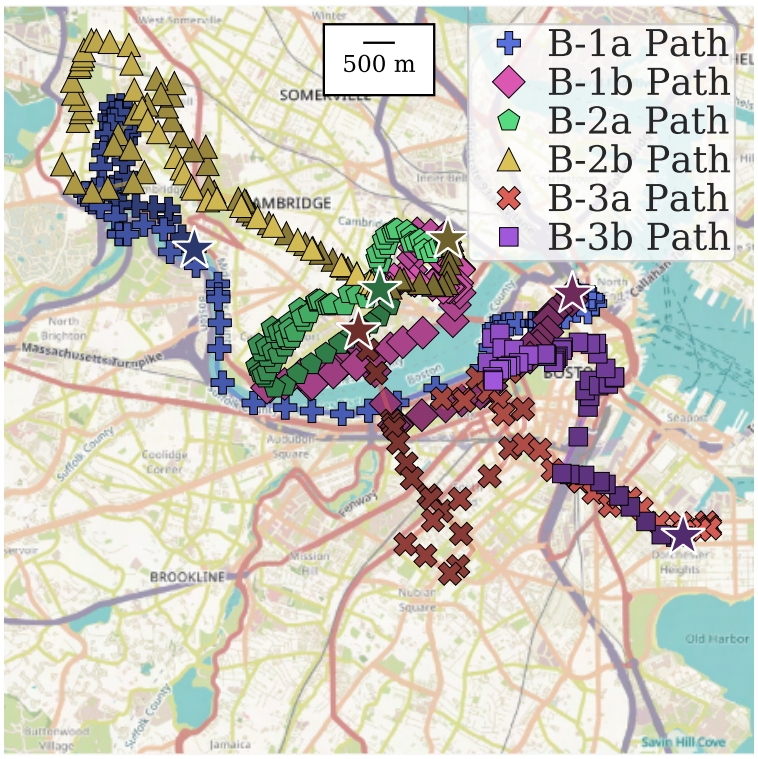
weights. This property has been shown to result in poor particle filter estimates [38]. ReWAG* instead uses systematic resampling that improves resampling quality and hence improves particle filter estimates over multinomial resampling [38].

ReWAG* also incorporates effective sample size (ESS) [80], a metric to control when resampling is performed. Resampling should only be done when new measurements are incorporated into the particle filter, because resampling without new measurements wastes computation and leads to sample impoverishment [52]. Sample impoverishment occurs when the particle filter ends up with a large portion of weight is allocated to a relatively small number of particles, leaving the majority of particles with a very small weight. In the bootstrap particle filter, which was previously used in ReWAG [28] and WAG [27], resampling was done at each time step regardless of whether new measurements were received. Extended testing on data in Boston revealed particle degeneracy issues with resampling done in [28]. Particle degeneracy is a similar and related issue to sample impoverishment; it occurs when low weight particles are never resampled and are lost, leading to less diversity in particles across the search space. ReWAG* checks ESS after each weighting step and only performs resampling when ESS drops below a threshold.

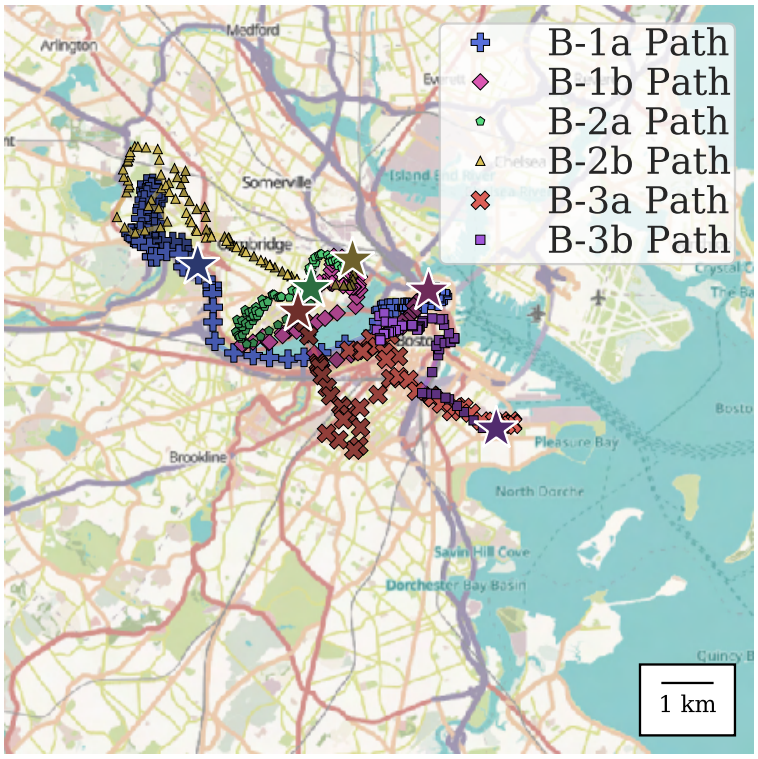
5.3 Results: Large-scale Hardware Experiment in Boston

5.3.1 Overview

The following sections detail the results of implementing ReWAG* with systematic resampling and training data augmentation and their relative effects compared to ReWAG without these improvements. Section 5.3.2 describes the experiments on data collected in fall 2022. Section 5.3.3 describes the experiments conducted in winter 2023.



(a) Close-up view of Boston paths.



(b) Boston paths shown within full search area.

Figure 5-5: Ground-truth paths in the Boston test area. Increasing brightness of path indicates passing of time. Start marked with stars.

(a) Image from November 11th.

(b) Image from November 15th.

(c) Image from November 21st.

(d) Image from November 21st.

Figure 5-6: Examples of images taken in November 2022.

5.3.2 Fall

The data in this section was collected between November 11th, 2022 and November 21st, 2022 and is characterized in Table 5.1 with paths shown in Fig. 5-5. Images from fall are challenging in part because of the different lighting conditions that occur. November in the Boston area is often cloudy or overcast, much more so than the summer months. This causes darker images as shown in Fig. 5-6(a) and Fig. 5-6(b). The tilt of the Earth on its axis in the fall also causes the sun to be lower on the horizon, casting longer rays with a more golden appearance as shown in Fig. 5-6(c) and 5-6(d).

The particle filter for this data was initialized with Gaussian distributions, centered 1.3 km from the true initial location (standard deviation of 500 m) for B-1b, B-2a, B-2b, B-3a and B-3b, which is the same initialization error as the Chicago

Table 5.1: Test Path Characteristics

Name	Date	Time of Day	Weather	Location	Length
B-1a	11/11/22	Afternoon	Cloudy	Cambridge & Boston	15 km
B-1b	11/11/22	Afternoon	Cloudy	Boston	13 km
B-2a	11/15/22	Morning	Sunny	Cambridge	10 km
B-2b	11/15/22	Morning	Partly Cloudy	Cambridge	17 km
B-3a	11/23/22	Afternoon	Sunny	Boston	14 km
B-3b	11/23/22	Afternoon	Sunny	Boston	8 km
wB-1a	3/6/23	Morning	Sunny	Cambridge	15 km
wB-1b	3/6/23	Morning	Sunny	Boston	17 km

experiments in Chapter 3 and Chapter 4 but with lower standard deviation due to the challenging generalization conditions of the Boston data. The particle filter was initialized with a Gaussian distribution centered 800 m from the true initial location (standard deviation of 150 m) for B-1a; this path requires lower initialization error because of the combination of dark conditions and long stretches near a river.

Localization on the images from this Boston dataset is a significant challenge for several reasons: ReWAG* was not trained on any images of Boston, the Boston images have a 72° FOV while ReWAG* was trained on 90° FOV images, the lighting conditions vary significantly from training images, and the Boston images were taken in the fall, at a different time of year than most training images, which were taken in the summer. Despite these challenges, ReWAG* was able to localize to roughly the same level of accuracy in Boston as ReWAG did in the simulated Chicago data. ReWAG*'s results with both of these improvements can be seen in Table 5.2, where ReWAG* outperforms TransGeo, the vision transformer baseline [115], on every path. TransGeo is used as a baseline because of vision transformers' impressive performance on cross-view geolocalization tasks and generalization on image tasks.

Resampling ablation. Fig. 5-7 shows how, with the original multinomial resampling strategy from [28], ReWAG* fails to accurately converge on the challenging path along the Charles River. Multinomial resampling at every timestep, with no use of ESS to control resampling, yields a final error of over 218 m. Systematic resam-

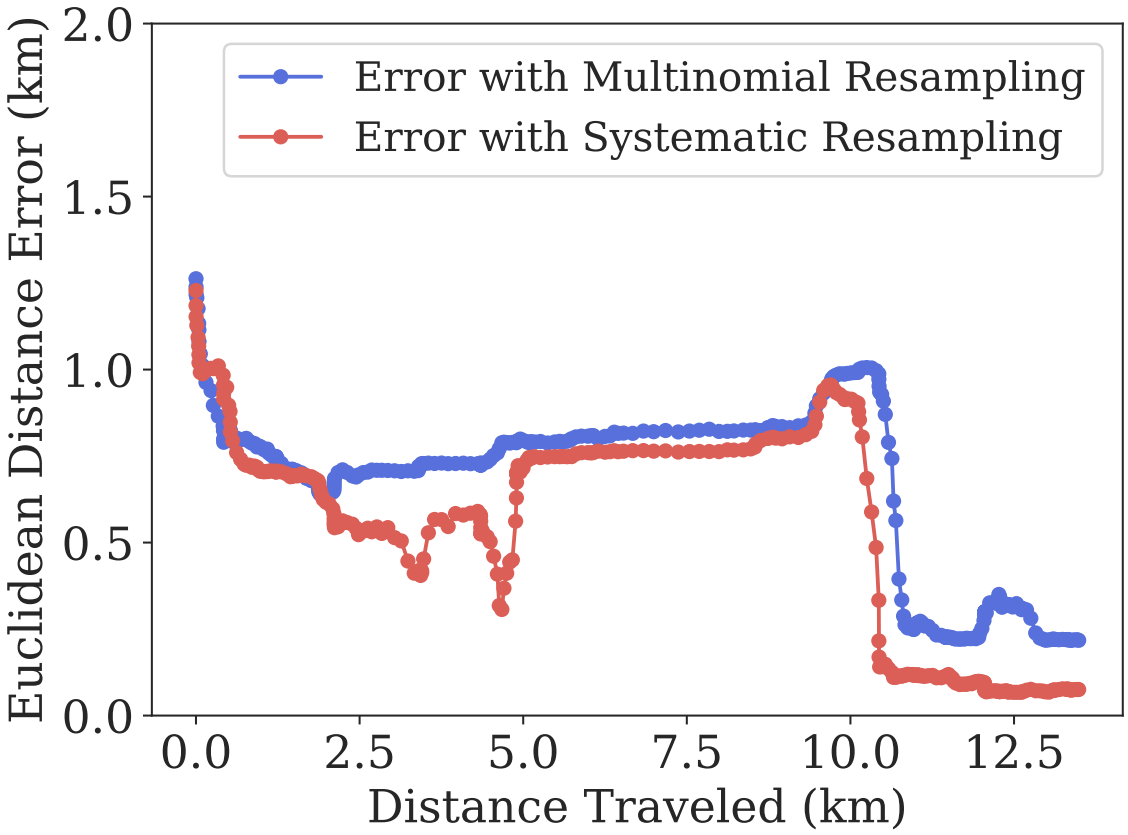


Figure 5-7: Comparison of ReWAG* with multinomial resampling and with systematic resampling on simulation of B-1b test path. Multinomial resampling does not converge, while systematic resampling converges after 10 km.

pling with an 0.98 ESS was able to achieve a final error of 75 m and convergence after 220 timesteps, demonstrating ReWAG*'s improved performance. Table 5.3 summarizes the performance with multinomial resampling at every timestep compared to ReWAG* with systematic resampling only when ESS drops below 0.98. While multinomial resampling decreases final estimation error and time to convergence in some paths, in others it fails to converge at all. ReWAG* with systematic sampling converges on every test path and hence yields more reliable performance.

Data augmentation ablation. There were variable lighting conditions while collecting data in Boston, including a dark, overcast day with diffuse, low lighting (in paths B-1a and B-1b) and a partly cloudy period with lower lighting (B-2b). These conditions were not seen frequently in the training dataset. To remedy this, ReWAG*

Table 5.2: Boston Experiment Results

Metric and System Type		B-1a	B-1b	B-2a	B-2b	B-3a	B-3b
Final Error (m)	Baseline	864	1329	1891	1884	1447	1690
	ReWAG*	40	75	151	105	222	62
Conv. Time (t-steps)	Baseline	-	-	-	-	-	-
	ReWAG*	329	220	347	247	166	128

Table 5.3: ReWAG* Ablation

Metric and System Type		B-1a	B-1b	B-2a	B-2b	B-3a	B-3b
Final Error (m)	No Augmentation	79	853	131	2206	125	61
	Multinomial	72	218	75	167	60	223
	ReWAG*	40	75	151	105	222	62
Conv. Time (t-steps)	No Augmentation	302	-	382	-	168	87
	Multinomial	254	-	347	-	136	-
	ReWAG*	329	220	347	247	166	128

incorporates Fancy PCA [51] data augmentation into ReWAG*'s training to improve the Siamese network's robustness to brightness changes. The PCA variable α was drawn from a Gaussian with mean 0 and standard deviation of 1000 to produce more drastic lighting differences. The results of training with Fancy PCA can be seen in Fig. 5-8, where ReWAG without Fancy PCA does not converge on the B-1b cloudy day and ReWAG with Fancy PCA does converge to 75 m of error. Table 5.3 summarizes the performance with and without Fancy PCA training data augmentation. Data augmentation reduces final estimation error in all but the B-3b path, and especially reduces estimation error on low light paths. Data augmentation may not reduce estimation error on the B-3b path because this path was well-lit and hence may not have been affected by lighting challenges.

5.3.3 Winter

To further demonstrate ReWAG*'s generalization ability, a dataset of images was collected along a route driven on a sunny winter day, March 6th, 2023. There was no visible snow on the ground but trees were empty of leaves. This route closely follows the paths driven for B-1a (driven mostly in Cambridge) and B-1b (driven in

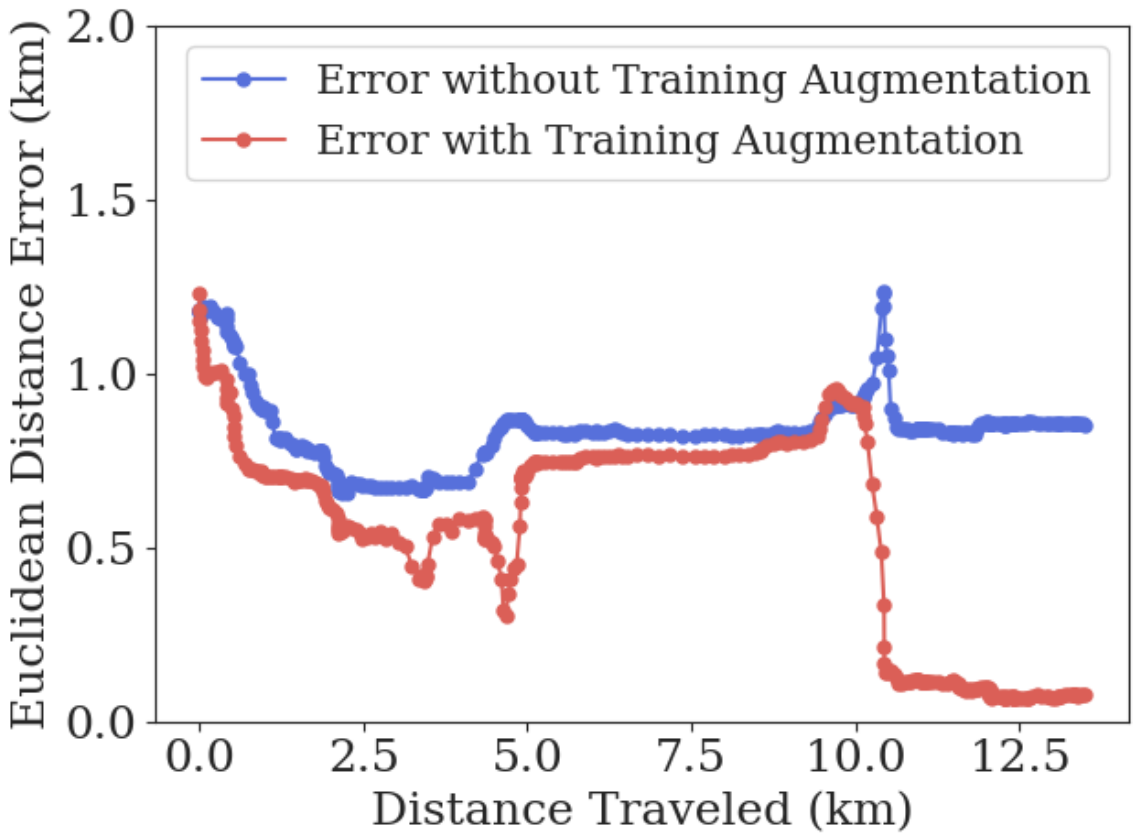
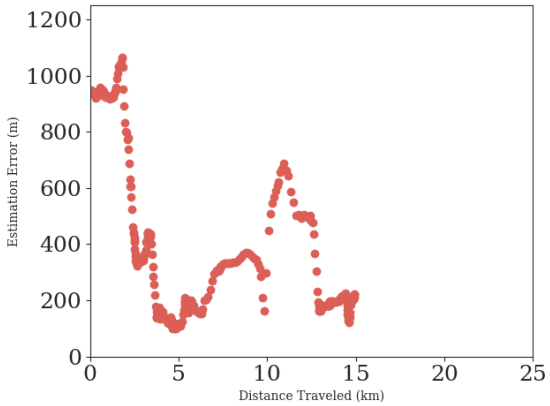
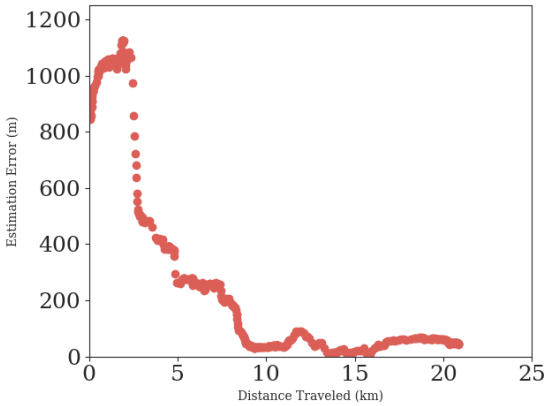


Figure 5-8: Comparison of ReWAG* performance on B-1b cloudy day with Fancy PCA training augmentation and without training augmentation.

Cambridge and Boston). It was collected with the same test apparatus as the fall paths. For the winter path, the particle filter was initialized with 800 m of error and a standard deviation of 150 m. The winter path B-1b requires more accurate initialization for convergence due to the appearance differences from the summer satellite imagery. The estimation error of path B-1a is shown in Fig. 5-9 and the estimation error of path B-1b is shown in Fig. 5-10. Both figures show that the estimation error is similar across both fall and winter paths, demonstrating ReWAG*'s seasonal invariance.

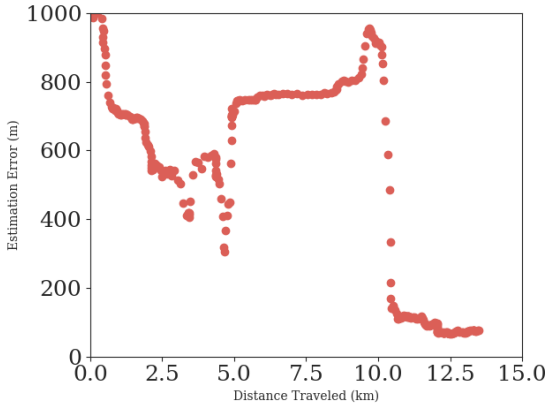


(a) Estimation error on fall B-1a

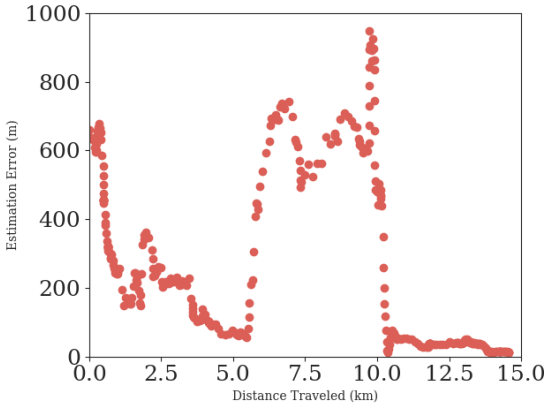


(b) Estimation error on winter B-1a

Figure 5-9: Example of consistent localization on fall and winter paths B-1a. The fall path begins approximately 5 km after the winter path starts.



(a) Estimation error on fall B-1b



(b) Estimation error on winter B-1b

Figure 5-10: Example of consistent localization on fall and winter paths B-1b. The winter path continues for approximately 2 km in Cambridge after the fall path.

5.4 Summary

This chapter described how ReWAG* performs wide-area cross-view geolocalization on challenging, realistic data collected with a moving vehicle in the Boston area. ReWAG* is trained with Fancy PCA image augmentation to introduce a greater variety of brightness variations in training data and hence improve localization performance on overly bright or overly dark images. ReWAG* also uses a particle filter with systematic resampling, which has favorable performance in instances that are

prone to particle degeneration. ReWAG*'s performance is demonstrated on ground images collected in Boston and Cambridge, Massachusetts in November 2022 and March 2023.

Although ReWAG* addresses some of the shortfalls of WAG and ReWAG, a gap remains in determining when the system has converged to an accurate position estimate, which is still difficult to discern without knowing the ground truth position. Chapter 6 details a method to check for convergence of particle filters used for localization. This method will be demonstrated to correspond with estimation error of ReWAG* as it localizes on the Boston data that has been described in this chapter.

Chapter 6

GKL Uncertainty Estimation

6.1 Introduction

A combination of particle standard deviation and estimation error has been used to determine convergence in previous chapters of this thesis; low standard deviation indicates that there is only one state that has high probability of being the true state and low estimation error indicates that the particle filter has found the correct location of the ground agent. However, both of these metrics are imperfect. Low standard deviation can occur even when the particles are clustered around a state that is far away from the true state, and estimation error can only be calculated with access to the ground truth state at each time step. This chapter will introduce a convergence metric that more reliably correlates with estimation error without requiring knowledge of the ground truth state.

6.2 Methods

6.2.1 Overview of Approach

The cross-view geolocalization systems described in this thesis could be used to provide global localization information to an autonomous system. An autonomous system may combine sensor information from multiple sources to produce the best lo-

calization, but to combine those sources of information most effectively it needs to know about the quality, uncertainty, or noise in each source. Cross-view geolocalization information is less helpful if its uncertainty cannot be determined. Since WAG, ReWAG and ReWAG* utilize particle filters to localize over time, they do not immediately provide accurate localization information after initialization. It takes a number of filter updates before the systems can converge and localize reliably with low estimation error. However, it is not possible to calculate the estimation error directly during deployment since ground truth location is not known. This leaves the problem of how to determine the convergence point, or when to trust the location outputs of the cross-view geolocalization systems. Uncertainty quantification is one way to determine when and how much to trust the system outputs.

This thesis envisions its cross-view geolocalization systems as self-contained modules that would be incorporated into larger navigation systems. Hence it focuses on estimating uncertainty of the particle filter localization output because that is the uncertainty that would be incorporated into a navigation system. There remains the possibility to estimate the uncertainty of each Siamese network output, using this uncertainty to improve the internal weighting of the particle filter. This thesis leaves Siamese network uncertainty estimation to future work.

This thesis proposes Gaussian Kullback-Leibler (GKL) divergence to quantify uncertainty of a cross-view geolocalization particle filter estimate by measuring the distance between the particle distribution and a Gaussian distribution. The KL divergence is estimated with a k-Nearest-Neighbor (kNN) probability density estimation, which reduces the computational requirements. GKL uncertainty corresponds to estimation error and hence can serve as a measure of particle filter convergence or uncertainty.

In the next sections the theory behind GKL uncertainty will be explained. Section 6.2.2 gives some background on Central Limit Theorems (CLTs). Section 6.2.3 details how GKL uncertainty is calculated.

6.2.2 Central Limit Theorems

Particle filter literature uses the term "convergence" to refer to the mean squared error, or the average squared difference between the particle states and the actual state, approaching zero [19]. In deployment of real systems, it is not possible to calculate the mean squared error to determine convergence because the true state is not known. Hence, this thesis uses the term convergence to refer to states where the particle locations have low standard deviation, regardless of whether that is close to the true state. If the particle states have low standard deviation and are close to the true state, then this thesis refers to that as true convergence. If the particle states have low standard deviation and are not close to the true state, then this thesis refers to that as false convergence.

Previous work has analyzed particle filter convergence error with Central Limit Theorems (CLTs), which provide measures of asymptotic variance [18, 25, 70]. The asymptotic variance provided by CLTs is helpful for comparing the efficiency of different types of particle filters, but CLTs also assert that a particle filter that reaches true convergence will have a Gaussian particle distribution.

Recent work [63, 64] proved a CLT for systematic auxiliary resampling, which is the particle filter structure used in ReWAG*. Hence, ReWAG* should converge to a Gaussian distribution as as it reaches true convergence and the mean squared error converges to zero. Particle filters often receive ambiguous measurements in the visual localization task due to perceptual aliasing. Perceptual aliasing occurs when different places generate similar visual outputs, and in the localization context perceptual aliasing can yields multiple potential states. If the particle distribution is multimodal and hence non-Gaussian, it follows from CLTs that the filter has not reached true convergence [63, 64]. Importantly, these CLTs do not claim that a Gaussian particle distribution indicates that a particle filter must have reached true convergence, only that a converged particle filter but have a Gaussian distribution. Hence, a non-Gaussian distribution indicates that a particle filter is not converged, but a Gaussian distribution does not guarantee that a particle filter is converged.

6.2.3 GKL Uncertainty

Since CLTs show that a particle filter with low mean squared error will have a Gaussian distribution, the difference between a Gaussian distribution and the particle distribution can measure the particle filter's convergence level. KL divergence (Eqn. 2.1) is a statistical measure of how one probability distribution differs from a reference probability distribution. Hence KL divergence between the particle distribution and a Gaussian distribution can be used to measure the particle filter's convergence. KL divergence as a test for normality has been demonstrated to have high power against non-normal alternatives [9]. This thesis refers to the proposed metric as Gaussian KL (GKL) uncertainty to avoid confusion with the KL divergence approach [97]. The algorithm is described in Alg. 2, with the values used for the parameters shown in Table 6.1. The value of 2 for d was determined by the dimension of the search space for this cross-view geolocalization problem (searching across latitude and longitude). The value of 30,000 for N was determined to be the minimum number of particles that reliably allowed the particle filter to converge. The value of 30,000 for M was selected to match N, but determined to have minimal effect on accuracy (demonstrated in Section 6.3). The Gaussian distribution that is used for comparison in GKL uncertainty has a mean and standard deviation that is the same as the particle distribution's. Figures 6-1 and 6-2 show examples of the mean and standard deviation obtained from two different particle distributions. GKL uncertainty is a measure of aleatoric uncertainty, or noise inherent to the ground images through phenomena like perceptual aliasing.

GKL uncertainty uses a k-nearest-neighbor (kNN) probability density estimation approach [77] to estimate KL divergence between the particle distribution and samples from a Gaussian distribution with the same mean and standard deviation as the particle distribution. kNN probability density estimation estimates the probability density at a point by calculating the distance from that point to its nearest neighbors. Ref. [77] estimates KL divergence between two continuous distributions without directly estimating the densities, hence improving computational efficiency. KL diver-

Algorithm 2 GKL

Require: $M > 0$, N particles $\mathbf{x} \in \mathbb{R}^d : X = [\mathbf{x}^j]_{j=1...N}$
  compute $\bar{\mathbf{x}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}^j$
  compute $\sigma^2 = \frac{1}{N-1}\sum_{j=1}^{N}(\mathbf{x}^j - \bar{\mathbf{x}})^2$
  for $j = 1$ to M **do**
    sample $\mathbf{g}^j \sim \mathcal{N}(\bar{\mathbf{x}}, \sigma^2)$
    add $[\mathbf{g}^j]$ to G
  end for
  build KD trees of G and X, T_G and T_X
  initialize $S = 0$
  for $j = 1$ to M **do**
    query T_G for $d_G^1(\mathbf{x}^j)$, distance from nearest neighbor of G to $\mathbf{x}^j$
    query T_X for $d_X^2(\mathbf{x}^j)$, distance from second nearest neighbor of X to $\mathbf{x}^j$
    compute $s_j = \log \frac{d_G^1(\mathbf{x}^j)}{d_X^2(\mathbf{x}^j)}$
    add $S = S + s_j$
  end for
  compute uncertainty, $U_{GKL} = -\frac{Sd}{M} + \log \frac{N}{M-1}$
  return U_{GKL}

Table 6.1: GKL Uncertainty Parameter Values

Parameter	Value
d	2
N	30,000
M	30,000

gence estimated by kNN probability density estimation for a continuous distribution P from a reference continuous distributed Q is calculated as follows:

$$\widehat{D}_k(P\|Q) = -\frac{d}{n}\sum_{i=1}^{n} \log \frac{r_{k_1}(\mathbf{x}_i)}{s_{k_2}(\mathbf{x}_i)} + \log \frac{m}{n-1} \tag{6.1}$$

where $\mathbf{x}_i$ is the vector of samples drawn from P, $\mathbf{x}'_i$ is the vector of samples drawn from Q, d is the dimension of the distributions P and Q, n is the number of samples drawn from P, m is the number of samples drawn from Q, $r_{k_1}(\mathbf{x}_i)$ is the Euclidean distance of the k_1^{th} nearest neighbor of $\mathbf{x}_i$ in $\mathbf{x}_i$, and $s_{k_2}(\mathbf{x}_i)$ is the Euclidean distance of the k_2^{th} nearest neighbor of $\mathbf{x}_i$ in $\mathbf{x}'_i$.

GKL uncertainty is, in essence, a quantification of how Gaussian a particle dis-

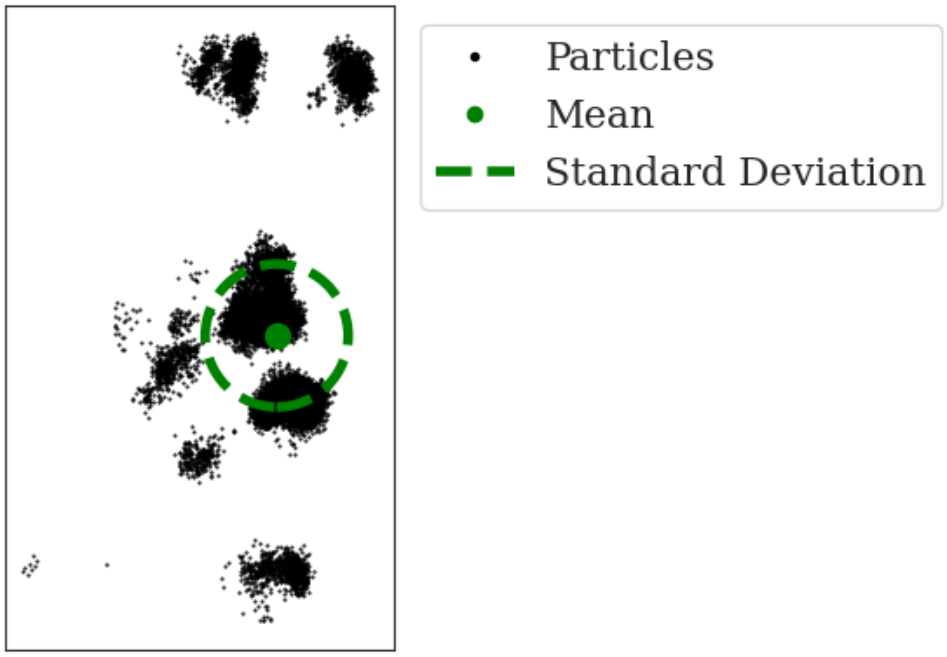


Figure 6-1: Mean and standard deviation are calculated from particle cloud at each time step. This figure shows an example of a particle filter that is not converged.

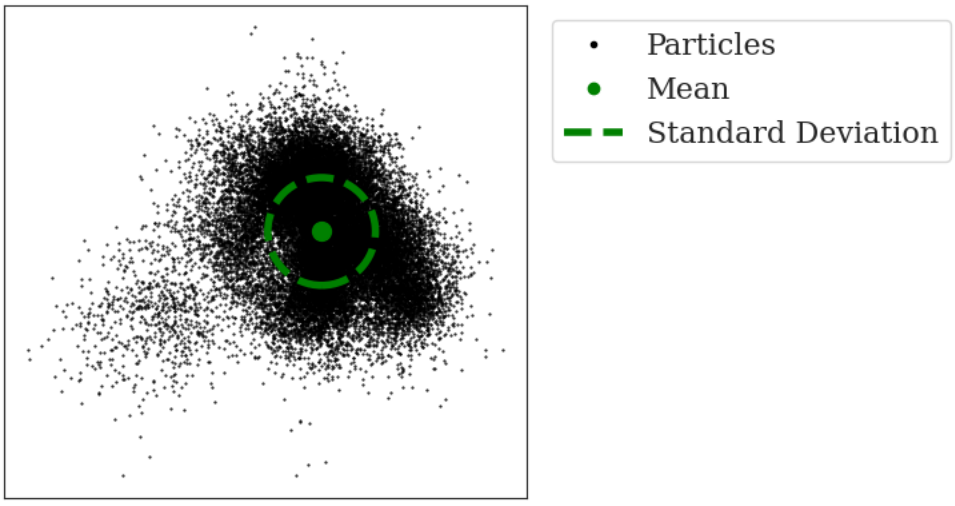


Figure 6-2: Mean and standard deviation are calculated from particle cloud at each time step. This figure shows an example of a particle filter that is converged.

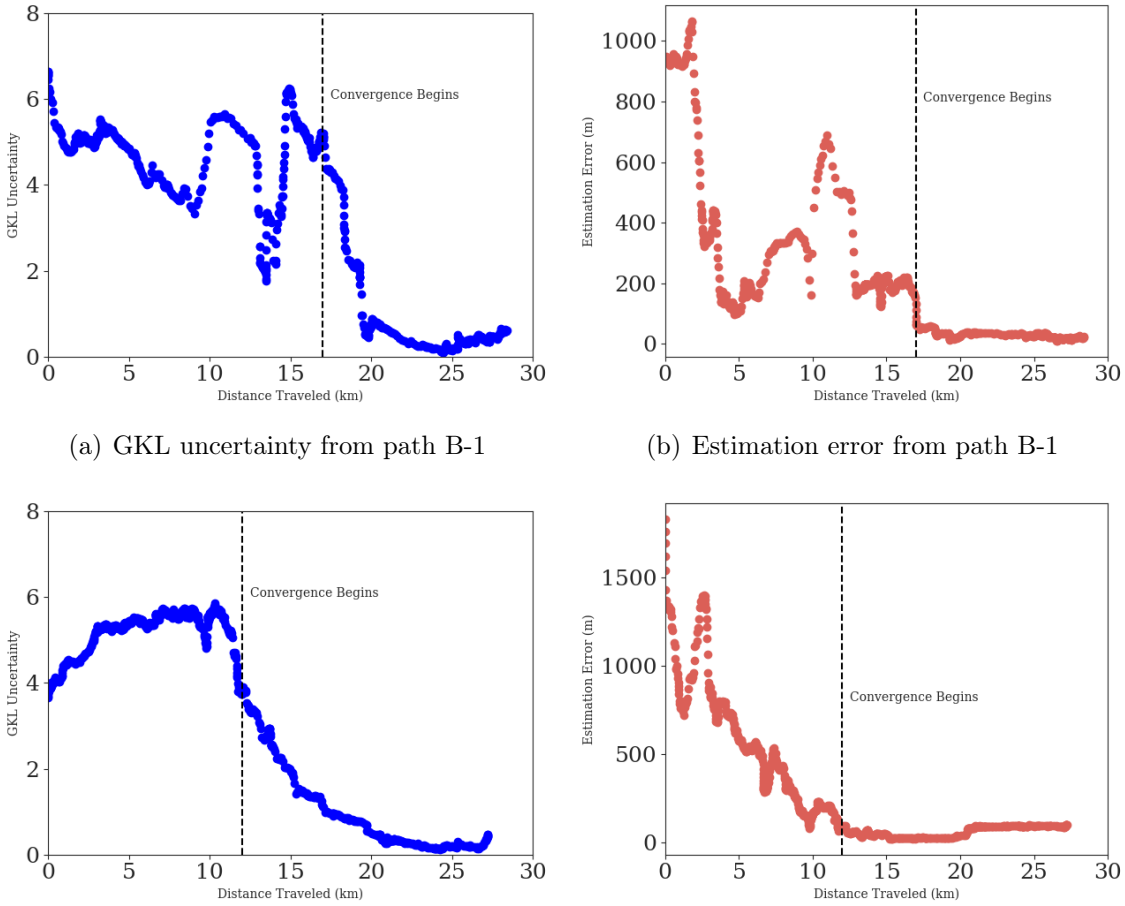
tribution is. As follows from CLTs, a non-Gaussian particle distribution cannot have reached true convergence. Hence, high GKL uncertainty indicates a highly non-Gaussian particle distribution which has not reached true convergence. In contrast, since a Gaussian particle distribution does not guarantee true convergence per CLTs, low GKL uncertainty does not guarantee true convergence. A low GKL uncertainty could be obtained for a false convergence. Hence, GKL uncertainty is more useful to identify false convergences than to guarantee true convergences.

The following section will show GKL uncertainty compared to estimation error on different paths from the Boston dataset. This thesis demonstrates a relationship between GKL uncertainty and true convergence on the full length Boston paths. In cases of true convergence, GKL uncertainty tends to decrease with a slight lag behind estimation error. GKL uncertainty is more prone to false positives than to false negatives; a high GKL uncertainty strongly indicates that the particle filter has not yet converged or is falsely converged while a low GKL uncertainty weakly indicates that the particle filter has reached true convergence. Essentially, GKL uncertainty has high recall but lower precision. As such, GKL uncertainty makes progress towards quantifying the uncertainty in ReWAG*'s output, but there remains space for additional work on uncertainty quantification.

6.3 Results

Figures 6-3 and 6-4 show the decrease in GKL uncertainty over distance traveled and its correlation with the decrease in estimation error for paths B-1 and B-2. Figure 6-4(a) shows a sharp spike in GKL uncertainty starting at around 20 km, and that directly corresponds with an increase in the estimation error due to an ambiguity that temporarily splits the particle cloud. GKL uncertainty can more accurately represent uncertainty caused by perceptual aliasing since it increases when the particles are multimodal and decreases when the particles are consistently clustering around one location.

GKL uncertainty is more suitable for measuring uncertainty than standard de-



(a) GKL uncertainty from path B-1

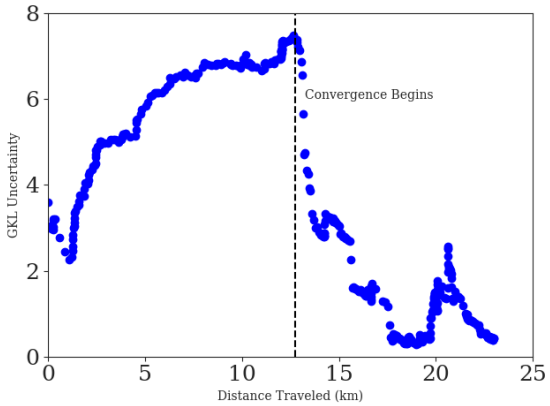
(b) Estimation error from path B-1

(c) GKL uncertainty from path B-2

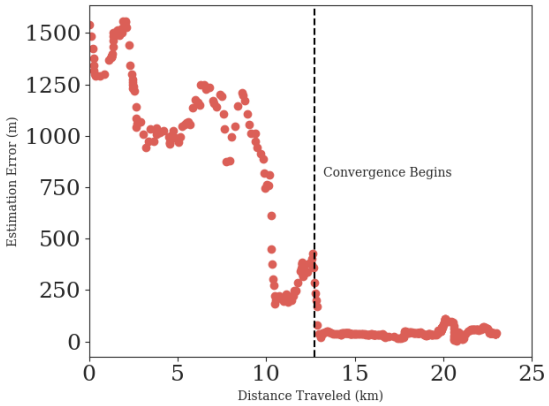
(d) Estimation error from path B-2

Figure 6-3: GKL uncertainty decreases as estimation error decreases in these examples of true convergence.

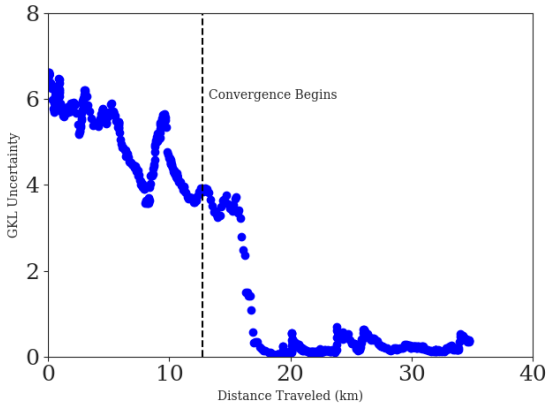
viation or effective sample size, which are often used as proxies for particle filter estimate uncertainty [26]. An example is shown in Fig. 6-5 comparing estimation error of a false convergence path in Fig. 6-5(a) with GKL uncertainty, standard deviation, and normalized effective sample size (effective sample size scaled by total number of particles). GKL uncertainty is the only metric that shows the uncertainty mostly increasing as the estimation error increases. Standard deviation decreases over time even as estimation error increases, and ESS stays relatively constant. Standard deviation and effective sample size do not yield vastly different uncertainties for a false convergence compared to a true convergence as shown in Fig. 6-6. GKL uncertainty enables identification of false convergence estimates.



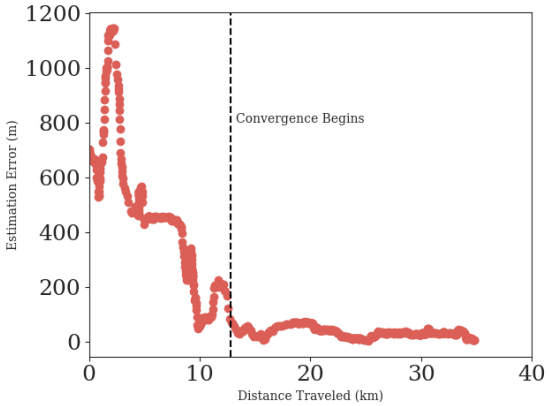
(a) GKL uncertainty from path B-3



(b) Estimation error from path B-3



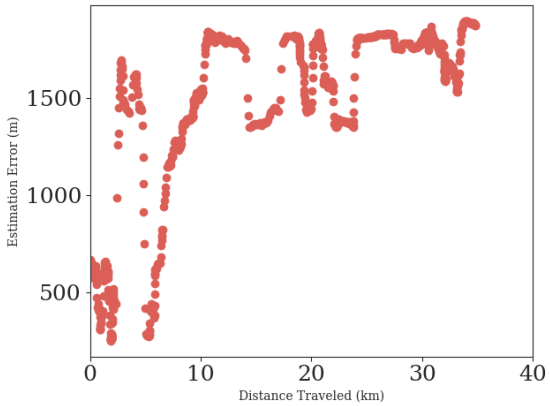
(c) GKL uncertainty from path wB-1



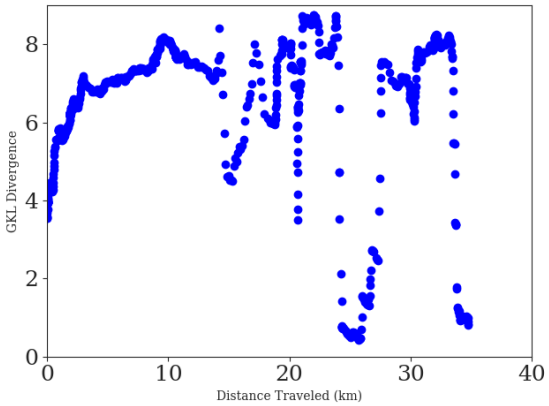
(d) Estimation error from path wB-1

Figure 6-4: GKL uncertainty decreases as estimation error decreases in these examples of true convergence.

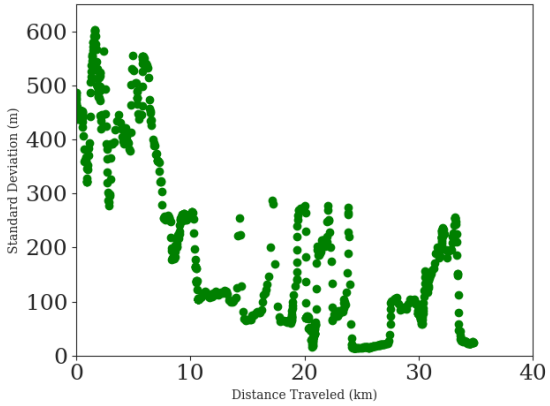
GKL uncertainty requires a relatively low computational effort and that effort can be scaled down with minimal effect on calculation accuracy due to its use of [77]'s k-nearest-neighbor (kNN) probability density estimation approach. For example, GKL uncertainty calculation takes 0.06 seconds with 30,000 samples of the Gaussian, and decreases to 0.007 seconds with 100 samples while resulting in similar GKL uncertainty values. Calculations were done on an NVIDIA GeForce RTX 3090. Calculation time can be decreased with less samples with a modest trade off of increased noise in the GKL uncertainty, as shown in Fig. 6-7.



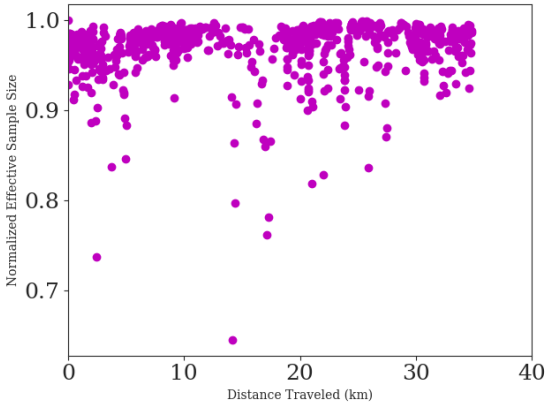
(a) Estimation error of false convergence



(b) GKL uncertainty increases as estimation error increases



(c) Standard deviation decreases even as estimation error increases

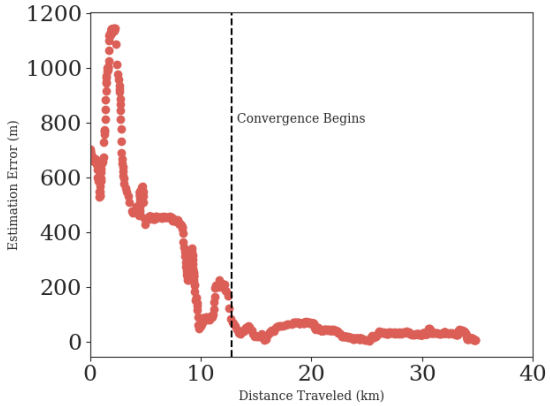


(d) Normalized effective sample size remains constant even as estimation error increases

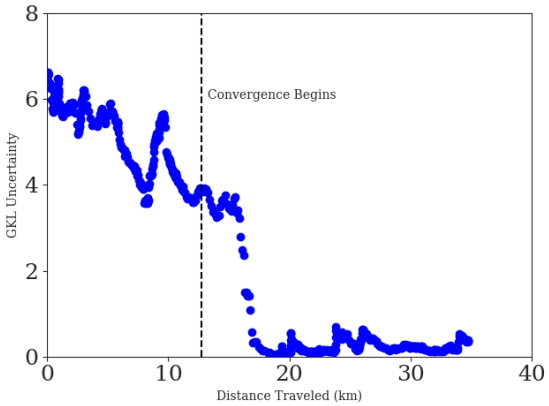
Figure 6-5: GKL uncertainty mostly outputs high uncertainty levels for a false convergence. Standard deviation decreases, suggesting decreased uncertainty, and normalized effective sample size remains constant, suggesting unchanging uncertainty. See Fig. 6-6 for true convergence.

6.4 Summary

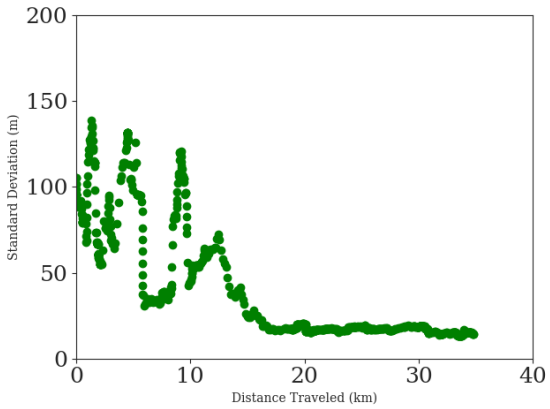
This chapter described how GKL uncertainty provides a metric for particle filter convergence and uncertainty. GKL uncertainty uses kNN probability density estimation to measure the distance between the particle distribution and a Gaussian distribution. GKL uncertainty is shown to closely track the estimation error obtained from various paths across different seasons of realistic Boston data. A navigation system can use GKL uncertainty to determine the point when ReWAG*'s localization esti-



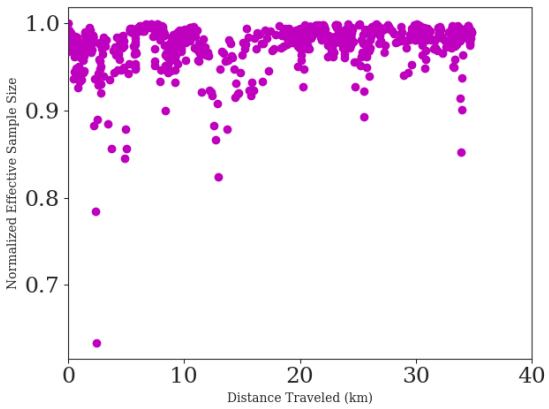
(a) Estimation error of true convergence



(b) GKL uncertainty decreases as estimation error decreases



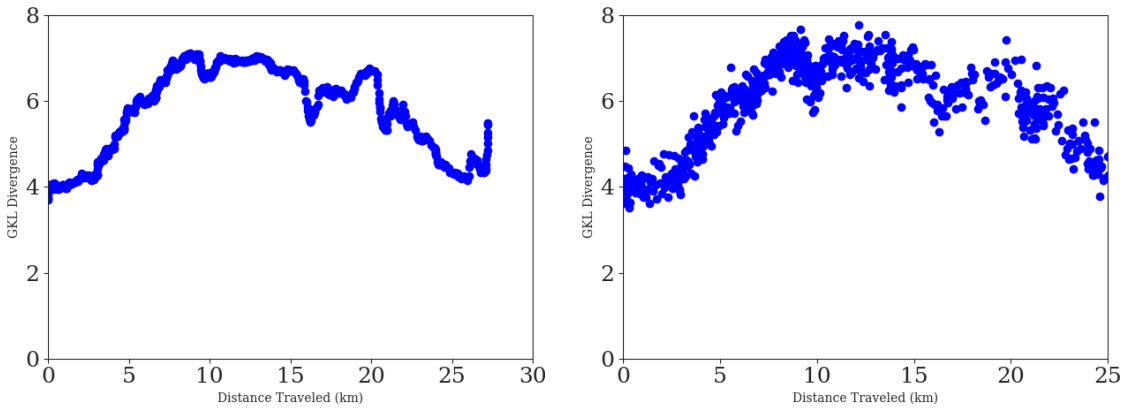
(c) Standard deviation decreases as estimation error decreases



(d) Normalized effective sample size remains constant even as estimation error decreases

Figure 6-6: GKL uncertainty accurately reflects low uncertainty levels for a true convergence. Standard deviation decreases, suggesting decreased uncertainty, and normalized effective sample size remains constant, suggesting unchanging uncertainty.

mate should begin to be incorporated to gain maximum information and calculate the most accurate state estimate.



(a) GKL uncertainty with 30,000 Gaussian samples

(b) GKL uncertainty with 100 Gaussian samples

Figure 6-7: GKL uncertainty result is similar with 30,000 samples and 100 samples but results in an order of magnitude reduction in computation.

Chapter 7

Conclusion

The core motivation for this work was that autonomous systems need GPS-denied localization to ensure robustness against GPS disruptions and adversarial attacks. Self-driving cars, delivery robots, military tactical awareness robots, and search and rescue robots all require reliable localization to enable safe operation. GPS-based localization is brittle and can be easily disrupted by systematic failures or targeted attacks. GPS-denied localization is a growing field, given a greater sense of urgency by government directives like the 2018 National Defense Authorization Act [1] and the 2020 executive order on Strengthening National Resilience Through Responsible Use of Positioning, Navigation and Timing Services [2], which mandate alternatives to GPS systems for critical infrastructure.

Cross-view geolocalization uses readily-available satellite imagery to localize with ground-view images and provides a backup or enhancement for GPS-based localization. Cross-view geolocalization can also be used to provide a check on GPS localization to detect spoofing. However, previous cross-view geolocalization systems have approached the problem from a computer vision perspective. Computer vision approaches often overlap with robotics approaches to many perception problems, but they typically make a different set of assumptions. Computer vision approaches often assume that the problem is being solved in post-processing, with access to significant computational resources and without strict time constraints. In robotics, there are often significant constraints on computational resources and localization algo-

rithms usually need to run in real-time. Although a few previous works have made some progress towards adapting cross-view geolocalization for robotics by integrating Siamese networks with particle filters [40, 47, 109, 117], there remained several key gaps.

The most significant literature gaps in existing cross-view geolocalization for robotics platforms have been performing localization across a large, city-scale search area, localizing with generic, non-specialized hardware, generalizing localization performance across diverse, challenging conditions, and quantifying the localization uncertainty of the system. This thesis has provided important insights and techniques for resolving these issues with existing cross-view geolocalization systems, resulting in a combined system suitable for GPS-denied automotive navigation.

First, the development of trinomial loss strengthens Siamese network performance on semi-positive image pairs, which enables superior computational time and storage scaling on larger search areas. Next, the creation of computationally efficient pose-aware embeddings improves cross-view geolocalization performance on cameras with smaller fields of view, thereby lifting the burdensome requirements of a panoramic camera. Additionally, the application of training data augmentation and the use of well-suited particle filter techniques aids in better generalization between training data and realistic, challenging deployment data. Finally, the proposed GKL divergence provides a first step towards quantifying the uncertainty of the particle filter output in a cross-view geolocalization system and a method to quantitatively assess the quality of the navigation solution.

Overall, this thesis demonstrates that cross-view geolocalization can be a valuable tool for autonomous navigation, and that changes in assumptions result in a cross-view geolocalization system that can localize faster, with fewer hardware constraints, across realistic imagery, and with estimates of uncertainty. Computer vision research on cross-view geolocalization has advanced quickly and succeeds at many challenging tasks, but the application of computer vision research towards robotics is more beneficial when the assumptions and constraints are truly representative of those of autonomous systems.

7.1 Future Work

This thesis has taken steps towards cross-view geolocalization for mobile autonomous systems, laying the groundwork that had to be completed before considering more complicated problems. However, there remain many potential avenues for further development. The environments trained and tested with in this thesis consisted primarily of urban and suburban scenery, which are more structured than rural or forested environments. As such, there may be different and novel techniques required to obtain similar performance in rural environments. This thesis has also focused on localization with no or very little GPS information, but further work could be done on the use of cross-view geolocalization to improve GPS estimates or to serve as an independent localization source when GPS fails. Furthermore, this thesis focused on localizing across a wide search area but did not achieve sub-GPS level accuracy. Additional work could be done to build a system that is capable of localizing from coarse to fine scales, or which is capable of converging faster so that it could be used to localize a pedestrian. Cross-view geolocalization could also be utilized in an active localization setting, as opposed to the passive localization studied in this thesis. Passive localization assumes that the localization system receives measurements without any ability to control the movement of the agent. Active localization allows the localization system to influence where the agent travels, directing the agent to move in ways that improve localization accuracy [12]. Additionally, the particle filter technique used in the cross-view geolocalization approaches presented in this thesis are fairly simple; additional performance boosts could be obtained with more advanced filtering techniques.

Developments in Siamese network architecture, training and loss functions could improve image matching performance, enabling faster localization convergence. Two architectures that seems particularly promising are the Vision Transformer [24] and diffusion networks [37]. Furthermore, improved image matching performance could allow uniform initialization across the search area, as opposed to this thesis' informed Gaussian localization. Within the particle filter, the probability distribution could be

represented more accurately with a better measurement model for learned similarity from a Siamese network. There exists promising work in the field of obtaining uncertainty estimates from neural networks; these works could be applied to generating a better measurement model with careful consideration of computational constraints or to reduce processing power by incorporating measurement from only the most confident ground images.

7.2 Ethical Impact

Work on cross-view geolocalization has some ethical implications in regards to privacy and surveillance that should be mentioned. In the age of social media when many people choose to share camera footage of their lives online to the public, cross-view geolocalization puts the privacy of people's locations at risk. Individual images and video footage can increasingly be localized with less existing context and fewer measurements required. Many social media personalities have begun to censor aspects of their lives such as the view out of their bedroom window or the scenes visible from their car as they drive to the store, which will become more and more necessary for those looking to preserve their privacy as this technology develops. Cross-view geolocalization could also be used for surveillance; it should be used carefully in this context. One could imagine cell phone footage being used as evidence of a person's location during a critical time frame. However, the current state of the art introduced by this thesis requires approximately 15 km of motion and as such these potential surveillance applications are in the far future. Nonetheless, it is important to consider the potential vulnerabilities of this technology in those cases; cross-view geolocalization data should be not taken as infallible as the technology is still developing.

Bibliography

[1] National timing and resilience security act. 2018. Public Law 115-282, Section 514.

[2] Strengthening national resilience through responsible use of positioning, navigation and timing services. pages 9359–9361, 2020. Executive Order No. 13905.

[3] EL Afraimovich, VV Demyanov, and G Ya Smolkov. The total failures of gps functioning caused by the powerful solar radio burst on december 13, 2006. *Earth, planets and space*, 61:637–641, 2009.

[4] Data Analytics and Resources Bureau. Road miles by municipality. Technical report, Massachusetts Department of Revenue Division of Local Services, 2023. url https://dlsgateway.dor.state.ma.us/reports/rdPage.aspx?rdReport=Socioeconomic.RoadMIles.

[5] Constantin-Octavian Andrei, Jan Johansson, Hannu Koivula, and Markku Poutanen. Signal performance analysis of the latest quartet of galileo satellites during the first operational year. In *2020 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–6, 2020.

[6] Constantin-Octavian Andrei, Sonja Lahtinen, Markku Poutanen, Hannu Koivula, and Jan Johansson. Galileo l10 satellites: Orbit, clock and signal-in-space performance analysis. *Sensors*, 21(5):1695, 2021.

[7] Jacqueline Ankenbauer, Kaveh Fathian, and Jonathan P How. View-invariant localization using semantic objects in changing environments. *arXiv preprint arXiv:2209.14426*, 2022.

[8] Jacqueline Ankenbauer, Parker C Lusk, and Jonathan P How. Global localization in unstructured environments using semantic object maps built from various viewpoints. *arXiv preprint arXiv:2303.04658*, 2023.

[9] Ikuo Arizono and Hiroshi Ohta. A test for normality based on kullback-leibler information. *The American Statistician*, 43(1):20–22, 1989.

[10] Mayank Bansal, Harpreet S. Sawhney, Hui Cheng, and Kostas Daniilidis. Geo-localization of street views with aerial image databases. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, page 1125–1128, New York, NY, USA, 2011. Association for Computing Machinery.

[11] Jane Bromley, James Bentz, Leon Bottou, Isabelle Guyon, Yann Lecun, Cliff Moore, Eduard Sackinger, and Rookpak Shah. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7:25, 08 1993.

[12] Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Active mobile robot localization. In *IJCAI*, pages 1346–1352. Citeseer, 1997.

[13] Sudong Cai, Yulan Guo, Salman Khan, Jiwei Hu, and Gongjian Wen. Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8390–8399, 2019.

[14] Rui Cao, Jiasong Zhu, Qing Li, Qian Zhang, Qingquan Li, Bozhi Liu, and Guoping Qiu. Learning spatial-aware cross-view embeddings for ground-to-aerial geolocalization. In *ICIG*, 2019.

[15] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3), 2010.

[16] Chao Chen, Xinhao Liu, Xuchu Xu, Yiming Li, Li Ding, Ruoyu Wang, and Chen Feng. Self-supervised visual place recognition by mining temporal and feature neighborhoods. *arXiv preprint arXiv:2208.09315*, 2022.

[17] Dapeng Chen, Zejian Yuan, Badong Chen, and Nanning Zheng. Similarity learning with spatial constraints for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1268–1277, 2016.

[18] Nicolas Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385 – 2411, 2004.

[19] Dan Crisan and Arnaud Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on signal processing*, 50(3):736–746, 2002.

[20] Youjing Cui and Shuzhi Sam Ge. Autonomous vehicle positioning with gps in urban canyon environments. *IEEE transactions on robotics and automation*, 19(1):15–25, 2003.

[21] Kelsey Davenport. Indian asat test raises space risks. *Arms Control Today*, 49(4):34–35, 2019.

[22] Petar M Djuric, Jayesh H Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F Bugallo, and Joaquin Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.

[23] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Shin-Fang Ch'ng, Thanh-Toan Do, and Ian Reid. Visual localization under appearance change: filtering approaches. *Neural Computing and Applications*, 33:7325–7338, 2021.

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[25] Randal Douc and Eric Moulines. Limit theorems for weighted samples with applications to sequential monte carlo methods. *The Annals of Statistics*, 36(5):2344–2376, 2008.

[26] Arnaud Doucet, Adam M Johansen, et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[27] 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 *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11102–11108, 2022.

[28] Lena M. Downes, Ted J. Steiner, Rebecca L. Russell, and Jonathan P. How. Wide-area geolocalization with a limited field of view camera. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[29] 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. *arXiv preprint arXiv:2308.07432*, 2023.

[30] Jos Elfring, Elena Torta, and René van de Molengraft. Particle filters: A hands-on tutorial. *Sensors*, 21(2), 2021.

[31] Per K Enge. The global positioning system: Signals, measurements, and performance. *International Journal of Wireless Information Networks*, 1:83–105, 1994.

[32] Masoud Faraki, Xiang Yu, Yi-Hsuan Tsai, Yumin Suh, and Manmohan Chandraker. Cross-domain similarity learning for face recognition in unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15292–15301, 2021.

[33] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[34] Zoubin Ghahramani. Unsupervised learning. In *Summer school on machine learning*, pages 72–112. Springer, 2003.

[35] Mohinder S Grewal, Angus P Andrews, and Chris G Bartone. *Global navigation satellite systems, inertial navigation, and integration*. John Wiley & Sons, 2020.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv e-prints*, page arXiv:1512.03385, December 2015.

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[38] Jeroen D. Hol, Thomas B. Schon, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82, 2006.

[39] Hui Hu and Na Wei. A study of gps jamming and anti-jamming. In *2009 2nd international conference on power electronics and intelligent transportation system (PEITS)*, volume 1, pages 388–391. IEEE, 2009.

[40] Sixing Hu and Gim Hee Lee. Image-based geo-localization using satellite imagery. *International J. of Computer Vision*, pages 1205–1219, 2020.

[41] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, March 2021.

[42] Nathan Jacobs, Scott Satkin, Nathaniel Roman, Richard Speyer, and Robert Pless. Geolocating static cameras. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–6, 2007.

[43] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.

[44] Krzysztof Jaskólski, Andrzej Felski, and Paweł Piskur. The compass error comparison of an onboard standard gyrocompass, fiber-optic gyrocompass (fog) and satellite compass. *Sensors*, 19(8):1942, 2019.

[45] Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya Gupta. To trust or not to trust a classifier. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5546–5557, Red Hook, NY, USA, 2018. Curran Associates Inc.

[46] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[47] Dong-Ki Kim and Matthew R. Walter. Satellite image-based localization via learned embeddings. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2073–2080, 2017.

[48] Orit Kliper-Gross, Tal Hassner, and Lior Wolf. One shot similarity metric learning for action recognition. In *Similarity-Based Pattern Recognition: First International Workshop, SIMBAD 2011, Venice, Italy, September 28-30, 2011. Proceedings 1*, pages 31–45. Springer, 2011.

[49] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies*, page 3–24, NLD, 2007. IOS Press.

[50] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[52] Chanin Kuptametee and Nattapol Aunsri. A review of resampling techniques in particle filtering framework. *Measurement*, 193:110836, 2022.

[53] Mingxiu Lin, Gang Xu, Xingning Ren, and Ke Xu. Cylindrical panoramic image stitching method based on multi-cameras. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 1091–1096, 2015.

[54] Tsung-Yi Lin, Serge Belongie, and James Hays. Cross-view image geolocalization. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2013.

[55] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5007–5015, 2015.

[56] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5617–5626, 2019.

[57] Peidong Liu, Marcel Geppert, Lionel Heng, Torsten Sattler, Andreas Geiger, and Marc Pollefeys. Towards robust visual odometry with a multi-camera system. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1154–1161, 2018.

[58] Elena López, Sergio García, Rafael Barea, Luis M Bergasa, Eduardo J Molinos, Roberto Arroyo, Eduardo Romera, and Samuel Pardo. A multi-sensorial simultaneous localization and mapping (slam) system for low-cost micro aerial vehicles in gps-denied environments. *Sensors*, 17(4):802, 2017.

[59] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *ieee transactions on robotics*, 32(1):1–19, 2015.

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

[61] Dominic Maggio, Marcus Abate, Jingnan Shi, Courtney Mario, and Luca Carlone. Loc-nerf: Monte carlo localization using neural radiance fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4018–4025. IEEE, 2023.

[62] Andreas Masselli, Richard Hanten, and Andreas Zell. Localization of unmanned aerial vehicles using terrain classification from aerial images. In *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13*, pages 831–842. Springer, 2016.

[63] Alessandro Mastrototaro and Jimmy Olsson. Adaptive online variance estimation in particle filters: the alvar estimator. *Statistics and Computing*, 33(4):1–26, 2023.

[64] Alessandro Mastrototaro, Jimmy Olsson, and Johan Alenlöv. Fast and numerically stable particle-based online additive smoothing: The adasmooth algorithm. *Journal of the American Statistical Association*, 0(0):1–12, 2022.

[65] 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.

[66] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE international conference on robotics and automation*, pages 1643–1649. IEEE, 2012.

[67] Niluthpol Chowdhury Mithun, Kshitij S. Minhas, Han-Pang Chiu, Taragay Oskiper, Mikhail Sizintsev, Supun Samarasekera, and Rakesh Kumar. Cross-view visual geo-localization for outdoor augmented reality. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 493–502, 2023.

[68] Marko Modsching, Ronny Kramer, and Klaus ten Hagen. Field trial on gps accuracy in a medium size city: The influence of built-up. In *3rd workshop on positioning, navigation and communication*, volume 2006, pages 209–218, 2006.

[69] Daniel Mas Montserrat, Qian Lin, Jan Allebach, and Edward J Delp. Training object detection and recognition cnn models using data augmentation. *Electronic Imaging*, 2017(10):27–36, 2017.

[70] P.D. Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Probability and Its Applications. Springer New York, 2004.

[71] Miguel Ángel Muñoz-Bañón, Jan-Hendrik Pauls, Haohao Hu, Christoph Stiller, Francisco A Candelas, and Fernando Torres. Robust self-tuning data association for geo-referencing using lane markings. *IEEE Robotics and Automation Letters*, 7(4):12339–12346, 2022.

[72] Achille Murangira, Christian Musso, and Igor Nikiforov. Particle filter divergence monitoring with application to terrain navigation. In *2012 15th International Conference on Information Fusion*, pages 794–801, 2012.

[73] Mohammad Norouzi, David J Fleet, and Russ R Salakhutdinov. Hamming distance metric learning. *Advances in neural information processing systems*, 25, 2012.

[74] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

[75] Sobhan Naderi Parizi, Alireza Tavakoli Targhi, Omid Aghazadeh, and Jan-Olof Eklundh. Reading street signs using a generic structured object detection and signature recognition approach. In *VISAPP (2)*, pages 346–355, 2009.

[76] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pages 4055–4064. PMLR, 2018.

[77] Fernando Perez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *2008 IEEE International Symposium on Information Theory*, pages 1666–1670, 2008.

[78] Sam Pullen, Grace Gao, Carmen Tedeschi, and John Warburton. The impact of uninformed rf interference on gbas and potential mitigations. In *Proceedings of the 2012 International Technical Meeting of The Institute of Navigation*, pages 780–789, 2012.

[79] Maria Isabel Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43(46):3736–3741, 2004.

[80] Branko Ristic, Sanjeev Arulampalam, and Neil J. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, MA, 2004.

[81] R. Rodrigues and M. Tani. Are these from the same place? seeing the unseen in cross-view image geo-localization. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3752–3760, Los Alamitos, CA, USA, jan 2021. IEEE Computer Society.

[82] 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.

[83] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[84] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[85] Turgay Senlet, Tarek El-Gaaly, and Ahmed Elgammal. Hierarchical semantic hashing: Visual localization from buildings on maps. In *2014 22nd International Conference on Pattern Recognition*, pages 2990–2995, 2014.

[86] 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.

[87] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. In *NeurIPS*, 2019.

[88] 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.

[89] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[90] Ted J. Steiner, Robert D. Truax, and Kristoffer Frey. A vision-aided inertial navigation system for agile high-speed flight in unmapped environments. In *2017 IEEE Aerospace Conference*, pages 1–10, 2017.

[91] Erik Stenborg, Carl Toft, and Lars Hammarstrand. Long-term visual localization using semantically segmented images. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6484–6490. IEEE, 2018.

[92] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[93] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, Cambridge, Mass., 2005.

[94] Yicong Tian, Chen Chen, and Mubarak Shah. Cross-view image matching for geo-localization in urban environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1998–2006, 2017.

[95] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. On the requirements for successful gps spoofing attacks. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 75–86, 2011.

[96] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, Fredrik Kahl, and Torsten Sattler. Long-term visual localization revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2074–2088, 2022.

[97] Begumhan Turgut and Richard P Martin. Restarting particle filters: an approach to improve the performance of dynamic indoor localization. In *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, pages 1–7. IEEE, 2009.

[98] Sabine Kastner Ungerleider and Leslie G. Mechanisms of visual attention in the human cortex. *Annual review of neuroscience*, 23(1):315–341, 2000.

[99] Vishal Thanvantri Vasudevan, Abhinav Sethy, and Alireza Roshan Ghias. Towards better confidence estimation for neural models. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7335–7339, 2019.

[100] Anirudh Viswanathan, Bernardo R. Pires, and Daniel Huber. Vision-based robot localization across seasons and in remote locations. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4815–4821, 2016.

[101] Anirudh Viswanathan, Bernardo Rodrigues Pires, and Daniel F. Huber. Vision based robot localization by ground to satellite matching in gps-denied situations. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 192–198, 2014.

[102] Nam N. Vo and James Hays. Localizing and orienting street views using overhead imagery. In *ECCV*, 2016.

[103] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter. *Kalman filtering and neural networks*, pages 221–280, 2001.

[104] Teng Wang, Ye Zhao, Jiawei Wang, Arun K. Somani, and Changyin Sun. Attention-based road registration for gps-denied uas navigation. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4):1788–1800, 2021.

[105] Tingyu Wang, Zhedong Zheng, Yaoqi Sun, Tat-Seng Chua, Yi Yang, and Chenggang Yan. Multiple-environment self-adaptive network for aerial-view geo-localization. *arXiv preprint arXiv:2204.08381*, 2022.

[106] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3961–3969, 2015.

[107] Piotr Wozniak and Dominik Ozog. Cross-domain indoor visual place recognition for mobile robot via generalization using style augmentation. *Sensors*, 23(13), 2023.

[108] Pengcheng Wu, Steven CH Hoi, Hao Xia, Peilin Zhao, Dayong Wang, and Chunyan Miao. Online multimodal deep similarity learning with application to image retrieval. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 153–162, 2013.

[109] 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*, 6:5921–5928, 2021.

[110] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[111] Hongji Yang, Xiufan Lu, and Yingying Zhu. Cross-view geo-localization with layer-to-layer transformer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29009–29020. Curran Associates, Inc., 2021.

[112] Dong Yi, Zhen Lei, Shengcai Liao, and S. Li. Deep metric learning for person re-identification. *2014 22nd International Conference on Pattern Recognition*, pages 34–39, 2014.

[113] Xiaohan Zhang, Waqas Sultani, and Safwan Wshah. Cross-view image sequence geo-localization. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2913–2922, 2023.

[114] Jianwei Zhao, Qiang Zhai, Pengbo Zhao, Rui Huang, and Hong Cheng. Covisual pattern-augmented generative transformer learning for automobile geolocalization. *Remote Sensing*, 15(9), 2023.

[115] 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.

[116] Sijie Zhu, Taojiannan Yang, and Chen Chen. Revisiting street-to-aerial view image geo-localization and orientation estimation. In *2021 Winter Conf. on Applications of Computer Vision*, pages 756–765, 01 2021.

[117] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5316–5325, 2021.

[118] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.

[119] Yingying Zhu, Hongji Yang, Yuxin Lu, and Qiang Huang. Simple, effective and general: A new backbone for cross-view image geo-localization. *arXiv preprint arXiv:2302.01572*, 2023.