Wide-Area Geolocalization with a Limited Field of View Camera in Challenging Urban Environments

Lena M. Downes\textsuperscript{1,2,3}, Ted J. Steiner\textsuperscript{2}, Rebecca L. Russell\textsuperscript{2} and Jonathan P. How\textsuperscript{1}

Abstract—Cross-view geolocalization, a supplement or replacement for GPS, localizes an agent within a search area by matching ground-view images to overhead images. Significant progress has been made assuming a panoramic ground camera. Panoramic cameras’ high complexity and cost make non-panoramic cameras more widely applicable, but also more challenging since they yield less scene overlap between ground and overhead images. This paper presents Restricted FOV Wide-Area Geolocalization (ReWAG), a cross-view geolocalization approach that combines a neural network and particle filter to globally localize a mobile agent with only odometry and a non-panoramic camera. ReWAG creates pose-aware embeddings and provides a strategy to incorporate particle pose into the Siamese network, improving localization accuracy by a factor of 100 compared to a vision transformer baseline. This extended work also presents ReWAG\textsuperscript{*}, which improves upon ReWAG’s generalization ability in previously unseen environments. ReWAG\textsuperscript{*} repeatedly converges accurately on a dataset of images we have collected in Boston with a 72° field of view (FOV) camera, a location and FOV that ReWAG\textsuperscript{*} was not trained on.

I. INTRODUCTION

GPS is an external system for localization that is susceptible to failure through jamming, spoofing, and signal dropout due to dense foliage or urban canyons. Cross-view geolocalization [1]–[5] is a localization method that only requires images from a ground-view camera and preexisting overhead imagery, with or without 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). Satellite imagery at some resolution is widely available for most of the planet, even in forests and urban areas where GPS signals can be weaker.

Cross-view geolocalization with ground and overhead images is challenging due to the wide difference in viewpoints. Many existing works [5]–[8] rely upon the use of a panoramic ground camera because it effectively decreases the problem dimensionality by reducing the impact of heading. Although a panoramic ground camera can have different headings, the heading only affects the alignment of the image, not 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. As a result, few real-world systems can benefit from panoramic-based localization. Widespread adoption of cross-view geolocalization technology will require its applicability to platforms without panoramic imaging capabilities.

Most recent work on cross-view geolocalization takes a deep learning approach to the problem by using Siamese networks [3]–[7], [9]. 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 integrating their measurements over time with a particle filter [4]–[6], [10], [11]. However, existing particle filter geolocalization works are highly constrained—requiring some level of GPS data [6], [10], 180° (or greater) FOV of the ground camera [4]–[6], [10], [11], perfect initial location knowledge [5], or a search area of less than 2.5 km\textsuperscript{2} [4], [5]. These additional constraints are needed because the algorithms cannot efficiently geolocalize a limited FOV camera across a large search area. Localization with a limited FOV camera increases both the difficulty of the problem and the computational requirements.

Our contribution. Our approach, Restricted FOV Wide-Area Geolocalization (ReWAG), builds upon our previous work in [11] to enable efficient wide-area geolocalization with a restricted FOV camera through two key changes. The first
change is a computationally efficient method for matching limited FOV ground images to satellite images by appending relative pose information to the intermediate network embedding before inputting it to the Spatial Aware Feature Aggregator (SAFA) [2]. The second change is the dual incorporation of relative pose into both the Siamese network and the particle filter, which enables the probability distribution to be modeled more accurately. The only additional information these changes necessitate at runtime is a noisy heading from a compass, and they produce a cross-view geolocalization system that is capable of localizing across city-scale search areas using a 90° FOV or less camera. In summary, in this paper we demonstrate the following contributions for restricted FOV cross-view localization:

1) Efficient generation of pose-aware embeddings that generalize to unseen environments;
2) A particle filter system that more accurately models the probability distribution, resulting in lower average and final localization error;
3) Faster particle filter convergence than a ViT baseline [12]; and
4) A challenging new dataset for testing of urban cross-view geolocalization in Boston.

Differences from conference paper. This evolved paper extends our previous work [13] in three main ways. The first being that we have collected a dataset of images and GPS tags in Boston that were used to test ReWAG* and which is publicly available at https://doi.org/10.5281/zenodo.7818704. Testing on this dataset initiated the second aspect of the extension—changes in our particle filter resampling scheme. Namely, changing from multinomial resampling [14] at each time step to systematic resampling [15] when effective sample size [16] dips below a threshold. Testing on this Boston dataset also motivated the third aspect of the extension—incorporation of training data augmentations to improve performance on ground images with variable brightness. We retrained the Siamese network with ground image augmentations generated from Fancy PCA [17].

II. RELATED WORKS

Ground-to-aerial cross-view geolocalization. Cross-view geolocalization derives from previous work in the areas of scene recognition and image retrieval. Ground-to-aerial cross-view geolocalization pushes previous work to higher levels of difficulty due to the vast difference in viewpoints between ground and aerial images. Previous works have attempted to solve this problem with hand-crafted features and traditional computer vision techniques [8], [18]–[20], but recent works [2]–[5], [7], [9], [21]–[26] have mostly applied deep learning in the form of Siamese networks [27]. In recent years, recall at top-1 has been steadily rising, but most works [5]–[8] focus on panoramic ground images and report high recall at top-1 for these panoramic ground images. Recall at top-1 for limited FOV ground images is significantly lower than that for panoramic ground images. However, limited FOV cameras are much more common than panoramic cameras in practice.

Orientation-aware cross-view geolocalization. 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 appending orientation maps to images to be input to the Siamese network [7], by using Dynamic Similarity Matching (DSM) to calculate the correlation between ground and satellite images [28], or by jointly embedding the full satellite image as well as the satellite image portion that is visible in the limited FOV ground image [29]. Instead of jointly determining the most highly matching satellite image and the orientation, [30] assumes that the satellite image has already been determined, and they then use pose optimization to estimate the pose within that satellite image. 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.

Orientation-blind cross-view geolocalization. More recent works tend to treat orientation as an aspect of the problem that can be solved at the last step [9], or do not directly encode or estimate it at all [2], [12]. In [9] orientation-invariant embedding schemes are learned through a combination of global mining, binomial loss, and training data augmentation with random rotations. SAFA [2] is an attention mechanism that helps the network to learn image descriptors regardless of the large viewpoint difference between ground and satellite views. TransGeo [12] 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 can 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.

Particle filter implementations. Some previous works have combined image retrieval with particle filters to enable cross-view geolocalization over time as an agent moves through a search area [4], [5], [8], [10], [11], [18]. Particle filters in these systems use random discrete particles to model a probability distribution of the agent location given odometry, satellite images of the search area, and ground images. However, existing works have localized with particle filters that use wide angle ground cameras [5], [8], [11], [18]. Although [4], [10] localize a ground agent with a 180° ground camera instead of a panoramic 360° camera, commonly available cameras like those found in cell phones have FOV of 90° or less, making much fewer features visible in their images. These existing particle filter cross-view geolocalization systems are not able to accurately localize with extremely limited FOV images.
III. METHODS

A. Overview of Approach

ReWAG builds upon WAG [11] to enable localization across a wide search area with restricted FOV ground images. ReWAG uses WAG’s strategies of creating a coarse satellite image database, generating embeddings with Siamese networks based on VGG-16 [31] and SAFA [2], and localizing over time with a particle filter (see Fig. 2). However, ReWAG differs from WAG in two major ways: first, ReWAG generates pose-aware image embeddings in a computationally efficient manner, and 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 generic 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, a probabilistic representation of the ground image’s similarity with different areas of the search area. Then, the particle filter receives odometry and measurements from the pseudo-similarity matrix to produce a location estimate at each time step. ReWAG* is our extension to ReWAG beyond [13], which has the same structure and training as ReWAG except for modifications to training data augmentation and particle filter resampling. Now, we describe in detail the changes implemented in ReWAG to enable limited FOV localization and the changes in ReWAG* that improve its performance on challenging, realistic datasets.

B. Pose-Aware Embeddings

We have developed a method to train the ground Siamese network to generate pose-aware embeddings in a computationally efficient manner while minimally modifying the architecture. Like WAG, our Siamese network architecture is derived from that of [6], which consists of a VGG-16 backbone and a SAFA module to increase the network’s spatial understanding. In ReWAG, the ground Siamese network is modified to append the particle pose to the intermediate embedding that is output by the VGG-16 backbone, as shown in Fig. 3. This intermediate embedding appended with the particle pose is then input to SAFA, which learns a spatial-aware representation of the ground image. In terms of total computation, the VGG-16 backbone comprises 93% and SAFA comprises 7% of the time to produce one pose-aware embedding. The computationally efficient benefit of this method comes from...
the ability to generate one base embedding for each ground image, and then being able to append any pose to the base embedding to efficiently generate a pose-aware embedding with SAFA. 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. Then, the much lighter-weight SAFA is the only part that needs to do inference for each particle. In contrast, [29] generates pose-aware embeddings through a joint global and local pipeline, which means that the aerial image embedding can only be computed once heading is known. Hence there is no ability to generate aerial embeddings a priori, increasing computation at runtime.

C. Training Data Augmentation

ReWAG* is additionally trained with Fancy PCA [17] data augmentations to improve its robustness to image brightness changes relative to ReWAG [13]. Training data augmentation is a method to artificially increase the size of a neural network training dataset by applying transformations to the existing data. Augmentations often improve neural network robustness to changes that should not affect the neural network output. For example, image classifiers may be trained with random crops and rotations applied to the images, since these transforms do not affect the content of the image and the classifiers should predict the same class regardless. Since cross-view geolocalization is sensitive to spatial alignment, 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 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. Fancy PCA produces brightness changes with natural appearances.
ReWAG is orientation-aware while needing only 7% of the computation time required to generate one ground embedding for each particle at each time step, since it takes advantage of a shared intermediate embedding for each ground image. 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. The incorporation of domain-knowledge has been shown to improve deep neural network performance [32]. Without pose-aware embeddings, the Siamese network has to learn where in the satellite image to look for matching features. With pose-aware embeddings, we tell the Siamese network where in the satellite image to look for matching features and hence simplify the learning task.

E. Particle Filter Resampling

The evolved ReWAG* detailed in this work also benefits from improvements to its particle filter resampling strategy relative to [13]. Particle filter resampling is done after the reweighting step of the particle filter. 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 [11], [13]. This approach has the benefits of being 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 poorer particle filter estimates [33]. In ReWAG*, we have instead implemented systematic resampling as it has been shown to improve resampling quality and hence improve particle filter estimates over multinomial resampling [33].

We have also incorporated into ReWAG* effective sample size (ESS) [34], a metric to turn the resampling function off and on. 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 [35]. In the bootstrap particle filter, which we previously used in ReWAG [13] and WAG [11], resampling is done at each time step regardless of whether new measurements have been received. Our extended testing on data in Boston revealed particle degeneracy issues when performing the multinomial resampling done in [13], causing us to instead implement systematic resampling and to use ESS to control resampling frequency.

IV. RESULTS

A. Experimental Setup

We use a version of the neural network architecture from [6] with the VGG-16 backbone and the SAFA module, modified to generate pose-aware embeddings. The satellite network architecture is unchanged. We train both ReWAG’s Siamese network and our comparison baseline, TransGeo [12], on the VIGOR dataset [6] with the ground images cropped to 90°.

We train the TransGeo baseline for 50 epochs with the training parameters described in [12], and ReWAG for 30 epochs with triplet loss [5]:

$$L_{\text{triplet}} = \log \left(1 + e^{-\alpha(d_{pos} - d_{neg})}\right)$$ (1)

with $\alpha$ loss parameter set to 10. Then we train ReWAG for 15 additional epochs with trinomial loss [11]:

$$L_t = \log \left(1 + e^{-\alpha_p(S_p - m_p)}\right) + \log \left(1 + e^{\alpha_n(S_n - m_n)}\right)$$

$$+ \log \left(1 + e^{-\alpha_{semi}(S_{semi} - m_{semi})}\right)$$

with the parameter values used in [11]. The particle filter has 30,000 particles and uses the Gaussian measurement model from [11].

B. Large-scale Simulation: Chicago

Experiment details. To demonstrate ReWAG’s performance relative to WAG [11], we perform limited FOV ground image localization experiments with the same simulated test paths from [11] but, instead of using panoramic ground images, we crop the ground images to 90° FOV. The true path is shown in Fig. 5 and is shown over our estimated location at each time step. 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 noise proportional to displacement. A satellite image database was generated by sampling the search area approximately every 60 meters into a 256 x 256 grid of non-overlapping satellite image tiles. This grid size maintains a similar image size and resolution that the network was trained on; satellite images of size 640 x 640 pixels with a resolution of approximately 0.1 m/pixel. We initialized the particle filter 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. We add 2% noise to the ground-truth odometry and 1% noise to the ground-truth heading at each time step.

We ran this experiment with ReWAG and a baseline that uses TransGeo [12], a vision transformer-based approach, for its Siamese network combined with the same particle filter as ReWAG.

Estimation error. Even with an initialization as far from the agent as Fig. 6(a), ReWAG has a final estimation error of 26 m (visible in Fig. 6(b)). This estimation error is only 5 meters greater than that which was achieved with 360° ground images in WAG. Fig. 7 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 locations of the particles in the particle filter. 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
Fig. 5. 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.

(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.

Fig. 6. ReWAG is able to accurately localize an agent to within 26 m 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.

Fig. 7. Particle filter estimation error from ReWAG compared to a TransGeo baseline. ReWAG has lower final and average error.

Fig. 8. Particle filter estimation convergence from ReWAG compared to a TransGeo baseline. ReWAG accurately converges, the baseline does not.

Table I

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

C. Small-scale Simulation: KITTI

We also demonstrate ReWAG’s localization performance on a simulated KITTI test path in Fig. 10. This experiment demonstrates the ability of ReWAG to localize with...
Fig. 9. 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 18 m of standard deviation.

Fig. 10. Ground-truth and estimated path in the KITTI test area. Increasing brightness of path indicates passing of time. Start marked with stars.

TABLE II

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

more accurate initialization across a smaller search area. We sample images and pose data from the residential “2011_0_30_drive_0028” path and reduce the FOV to 90° by cropping the KITTI images. Satellite images of the search area are obtained from the Google Static API; the search area is divided into a grid of 32 x 32 satellite images at zoom level 20. We initialize the particle filter with a Gaussian distribution centered roughly 80 m away from the true location, and after 34 ground images, ReWAG’s final estimation error is 12 m.

D. Large-scale Hardware Experiment: Boston

We have collected approximately 6 hours of driving data in Boston, MA (and the adjacent city of Cambridge, MA) to further test ReWAG. This Boston data initiated changes which resulted in our evolved ReWAG, ReWAG*. The data was collected between November 11th, 2022 and November 21st, 2022 and is characterized in Table III with paths shown in Fig. 11. We drove a Lincoln MKZ (see Fig. 12) 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 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 [36], 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. 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 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. We initialized the particle filter 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, and centered 800 m from the true initial location (standard deviation of 150 m) for B-1a.

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 only trained on 90° FOV images, the lighting conditions vary significantly from training images, and the Boston images were taken at a different time of year than most training images (Fall vs. 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.

Our first tests on the Boston data revealed two key weaknesses of our initial version of ReWAG. First, ReWAG suffered from particle degeneracy that became particularly apparent in the B-1b and B3-b 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. 13. 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. Our testing on these challenging paths prompted improvements in our particle filter resampling strategies for ReWAG*.

Secondly, ReWAG was sensitive to the brightness of the ground images it was tested on. The B-1a and B1-b test paths, which were collected on an overcast, cloudy day, also highlighted this sensitivity. Originally, the Siamese network was trained on satellite images from Google Maps and ground images from Google StreetView without any data augmentation. However, Google StreetView images are not randomly dispersed in different weather conditions, they are overwhelmingly taken on bright and sunny, or partially sunny days, (see Fig. 14). Previously published Siamese networks for cross-view geolocalization had not done data augmentation during training, so ReWAG [13] did not either. 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 training data augmentation into ReWAG*’s training.

ReWAG*’s results with both of these improvements can be seen in Table IV, where ReWAG* outperforms the vision transformer baseline [12] on every path.

**Resampling ablation.** Fig. 15 shows how, with the original multinomial resampling strategy from [13], ReWAG* fails to accurately converge on the challenging path along the Charles River. Multinomial resampling at every timestep (regardless of ESS) yields a final error of over 218 m. Systematic resampling with an 0.98 ESS was able to achieve a final error of 75 m and convergence after 220 timesteps. Table V summarizes the
### Table IV. 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 |
| ReW AG*                | 23   | 75   | 99   | 105  | 43   | 62   |
| Conv. Time (t-steps)   | 329  | 220  | 347  | 247  | 166  | 128  |

### Table V. ReW AG* 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   |
| ReW AG*                | 23   | 75   | 99   | 105  | 43   | 62   |
| Conv. Time (t-steps)   | 302  | -    | 382  | -    | 168  | 87   |
| ReW AG*                | 329  | 220  | 347  | 247  | 166  | 128  |

performance with multinomial resampling at every timestep compared to ReW AG* 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. ReW AG* with systematic sampling converges on every test path and hence yields more reliable performance.

**Data augmentation ablation.** We encountered variable lighting conditions while collecting our 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, we incorporated Fancy PCA [17] data augmentation into ReW AG*’s training to improve our Siamese network’s robustness to brightness changes. We used $\alpha = 1000$ in training to produce more drastic lighting differences. The results of training with Fancy PCA can be seen in Fig. 16, where ReW AG without Fancy PCA does not converge on the B-1b cloudy day and ReW AG with Fancy PCA does converge to 75 m of error. Table V summarizes the performance with, and without, Fancy PCA training data augmentation; showing that the data augmentation reduces final estimation error in all but the B-3b path, and especially reduces estimation error on low light paths.

### V. Conclusion

ReW AG redesigns the Siamese network-particle filter architecture for increased hardware flexibility, a key component of applying cross-view geolocalization to mobile robotics. Previous works have largely focused on geolocalization with ground cameras that have much wider FOVs than cameras that would be readily available on existing platforms. This work uses all available information to generate Siamese network embeddings and accurately reweight particles, hence enabling faster and more accurate particle filter convergence with limited FOV cameras.

ReW AG* generalizes from training on bright, 90° FOV images of Chicago to accurately localize across several hundred square kilometers of Boston with 72° FOV images with variable brightness. ReW AG also accurately localizes across several hundred square kilometers of Chicago in simulation, maintaining the same level of accuracy that [11] previously demonstrated with panoramic ground cameras. ReW AG maintains the same benefits of W AG in terms of reducing the size of the satellite image database required. Additionally, it has lower final error and faster convergence than the TransGeo baseline in the Chicago test area, which we attribute to its training with trinomial loss. It also successfully converges to a location estimate that is 12 m from the true location in the KITTI test path, which further demonstrates its ability to generalize to a city it was not trained on.

In the short term, future work on this topic includes improving computational speed, testing on real-time physical platforms, and further restriction of the FOV. Domain shift remains an open challenge in the field—localizing in areas with different semantic appearances than training image pairs, on satellite images that were taken in different seasons than the ground images, and in rural areas without as many identifiable landmarks as residential or urban areas.

In summary, ReW AG lifts the heavy hardware requirements that were inherent in other cross-view geolocalization systems and enables accurate localization in previously unseen environments using a camera with as narrow a FOV as 72°. It represents a step forward in making cross-view geolocalization...
a widespread tool that can be used on many platforms with different sets of hardware.

REFERENCES

[1] Y. Tian, C. Chen, and M. Shah, “Cross-view image matching for geo-localization in urban environments,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1998–2006.

[2] Y. Shi, L. Liu, X. Yu, and H. Li, “Spatial-aware feature aggregation for image based cross-view geo-localization,” in NeurIPS, 2019.

[3] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, “Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 8390–8399.

[4] D.-K. Kim and M. R. Walter, “Satellite image-based localization via learned embeddings,” 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 2073–2080, 2017.

[5] S. Hu and G. H. Lee, “Image-based geo-localization using satellite imagery,” International J. of Computer Vision, pp. 1205–1219, 2020. [Online]. Available: https://doi.org/10.1007/s11263-019-01186-0

[6] S. Zhu, T. Yang, and C. Chen, “Vigor: Cross-view image geo-localization beyond one-to-one retrieval,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5316–5325, 2021.

[7] L. Liu and H. Li, “Lending orientation to neural networks for cross-view geo-localization,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5617–5626, 2019.

[8] A. Viswanathan, B. R. Pires, and D. 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, pp. 192–197, 2014.

[9] S. Zhu, T. Yang, and C. Chen, “Revisiting street-to-aerial view image geo-localization and orientation estimation,” in 2021 Winter Conf. on Applications of Computer Vision, 01 2021, pp. 756–765.

[10] Z. Xia, O. Booij, M. Manfredi, and J. F. P. Kooij, “Cross-view matching for vehicle localization by learning geographically local representations,” IEEE Robotics and Automation Letters, vol. 6, pp. 5921–5928, 2021.

[11] L. M. Downes, D.-K. Kim, T. J. Steiner, and J. P. How, “City-wide street-to-satellite image geo-localization of a mobile ground agent,” in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 1102–1108.

[12] S. Zhu, M. Shah, and C. 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, 2022, pp. 1162–1171.

[13] L. M. Downes, T. J. Steiner, R. L. Russell, and J. P. How, “Wide-area geo-localization with a limited field of view camera,” in 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023.

[14] R. J. Tibshirani and B. Efron, “An introduction to the bootstrap,” Statistics and computing, vol. 57, no. 1, 1993.

[15] A. Doucet, N. De Freitas, N. J. Gordon, et al., “Sequential Monte Carlo methods in practice.” Springer, 2001, vol. 1, no. 2.

[16] J. S. Liu, “Metroized independent sampling with comparisons to rejection sampling and importance sampling,” Statistics and computing, vol. 13, no. 2, pp. 84–90, 2007.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[18] A. Viswanathan, B. R. Pires, and D. Huber, “Vision-based robot localization across seasons and in remote locations,” in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 4815–4821.

[19] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless, “Geolocating static cameras,” in 2007 IEEE 11th International Conference on Computer Vision, 2007, pp. 1–6.

[20] M. Bansal, H. S. Sawhney, H. Cheng, and K.Daniilidis, “Geo-localization of street views with aerial image databases,” in Proceedings of the 19th ACM International Conference on Multimedia, ser. MM ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 1125–1128. [Online]. Available: https://doi-org.libproxy.mit.edu/10.1145/2072298.2071954

[21] S. Workman, R. Souvenir, and N. Jacobs, “Wide-area image geolocalization with aerial reference imagery,” 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3961–3969, 2015.

[22] N. N. Vo and J. Hays, “Localizing and orienting street views using overhead imagery,” in ECCV, 2016.

[23] 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), Los Alamitos, CA, USA: IEEE Computer Society, jan 2021, pp. 3752–3760. [Online]. Available: https://doi.org/10.1109/WACV48360.2021.00380

[24] R. Cao, J. Zhu, Q. Li, Q. Zhang, Q. Li, B. Liu, and G. Qiu, “Learning spatial-aware cross-view embeddings for ground-to-aerial geo-localization,” in ICIG, 2019.

[25] T.-Y. Lin, S. Belongie, and J. Hays, “Cross-view image geolocalization,” in 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 891–898.

[26] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, “Learning deep representations for ground-to-aerial geolocalization,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5007–5015.

[27] J. Bromley, J. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Sackinger, and R. Shah, “Signature verification using a ‘siamese’ time delay neural network,” International Journal of Pattern Recognition and Artificial Intelligence, vol. 7, p. 25, 08 1993.

[28] Y. Shi, X. Yu, D. Campbell, and H. 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, 2020, pp. 4064–4072.

[29] R. Rodrigues and M. 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, 2022, pp. 3871–3879.

[30] Y. Shi and H. 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, 2022, pp. 17010–17020.

[31] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in International Conference on Learning Representations, 2015.

[32] T. Dash, S. Chitlangia, A. Ahuja, and A. Srinivasan, “A review of some techniques for inclusion of domain-knowledge into deep neural networks,” Scientific Reports, vol. 12, no. 1, p. 1040, Jan 2022. [Online]. Available: https://doi.org/10.1038/s41598-021-04590-0

[33] J. D. Hol, T. B. Schon, and F. Gustafsson, “On resampling algorithms for particle filters,” in 2006 IEEE Nonlinear Statistical Signal Processing Workshop, 2006, pp. 79–82.

[34] B. Ristic, S. Arulampalam, and N. J. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications. Boston, MA: Artech House, 2004.

[35] C. Kuptametee and N. Aunsri, “A review of resampling techniques in particle filtering framework,” Measurement, vol. 193, p. 110836, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224122001312

[36] T. J. Steiner, R. D. Truax, and K. Frey, “A vision-aided inertial navigation system for agile high-speed flight in unmapped environments: Distribution statement a: Approved for public release, distribution unlimited,” in 2017 IEEE Aerospace Conference, 2017, pp. 1–10.