2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

Panoramic Image-Based Aerial Localization using Synthetic Data via Photogrammetric Reconstruction

Danial Sufiyan, Ying Hong Pheh, Luke Soe Thura Win, Shane Kyi Hla Win, U-Xuan Tan
and Shaohui Foong, *Member, IEEE*

***Abstract*—To successfully adhere to flight plans, aerial vehicles must keep track of their location in 3D space, which is usually reliant on external references such as GNSS which are susceptible to interference. To develop self-reliant onboard positional localization, a workflow using 360-degree panoramic images in an image-based localization system using a Deep Convolutional Neural Network is proposed. 360-degree panoramic images have the advantage that they take into account visual information from all angles. Model performance is also enhanced by generating synthetic data from a 3D model of the region of interest created via photogrammetry techniques. The performances of different training configurations are compared, and the configuration with mixed real and synthetic data exhibits the highest performance, an approximately 10 to 15 percent improvement over using solely real data. Additional image augmentations also further reduce the localization error by 8 to 15 percent.**

I. Introduction

Aerial vehicles (both manned and unmanned) navigate via a suite of sensors, mainly the combination of an Inertial Measurement Unit (IMU) and the Global Navigation Satellite System (GNSS) for positional reference. However, with the vulnerability of satellite-based localization (GNSS) (or any other radio-wave based localization methods) to interference, be it natural (weather, multi-pathing from objects) or artificial (jamming and spoofing), there is a growing need for an absolute localization system in 3D space without reliance on an external system. A possible solution for a self-reliant onboard localization method is to use image-based methods.

The idea of using a single image for localization has been quite attractive, and has been studied in numerous works such as [1], and also [2], [3] and [4], which uses Convolutional Neural Networks (CNNs) as a feature extractor and additional hidden layers for pose regression. Captured images are also automatically labelled using a Structure-from-Motion (SfM)-based workflow. Also, PoseNet [2] was one of the first few works to propose using CNNs for visual localization to solve the kidnapped robot problem. A more portable version of PoseNet for mobile devices was proposed in [5]. In the aerial domain, the use of satellite imagery as reference has also been explored. The authors in [6] proposed a similar solution to PoseNet but used high altitude satellite imagery as the training data, which allowed large-scale absolute visual localization for aerial vehicles. The work in [7] involves the development of a cross-view geolocalization method with the aid of georeferenced satellite imagery.

With the increase in availability of 360-degree cameras, several works have leveraged on this, for example [8], which proposed using 360-degree cameras for indoor localization using feature matching. The authors in [9] proposed using the native 360 images from Google Maps as a dataset for localization. Another work in [10] used 360-images matched with satellite imagery for crosswalk localization for visually impaired pedestrians. Other uses of panoramic images combined deep learning include depth perception [11], as well as more specific cases such as fire detection [12]. In our work, we extend and explore the use of panoramic images for aerial localization. Most of the works mentioned above focus on ground-based datasets with rich urban features.

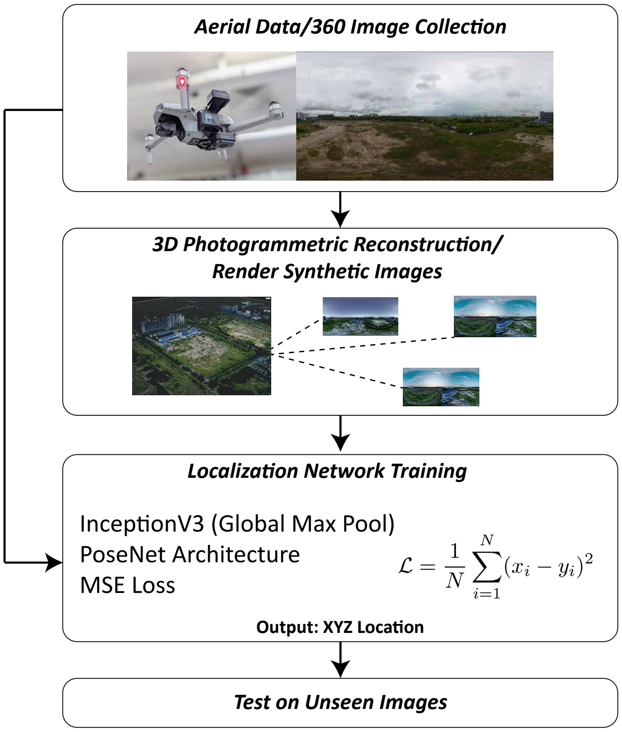

Aerial Data/360 Image Collection
3D Photogrammetric Reconstruction/
Render Synthetic Images
Localization Network Training
InceptionV3 (Global Max Pool)
PoseNet Architecture
MSE Loss
$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$
Output: XYZ Location
Test on Unseen Images

Fig. 1. Summary of the workflow proposed in this work.

Using deep neural networks for visual localization requires a rather dense sampling of the region of interest. Creating image datasets has always been a challenge, which is why there is an increasing trend towards using synthetic data to enhance model performance. In the realm of semantic segmentation, there are works such as [13] that prove its viability. Data generation for autonomous car use has also been explored in [14]. The notion of capturing reality and converting real-life environ-

978-1-6654-7633-1/23/$31.00 ©2023 IEEE

ments into digital twins has also become increasingly popular and accessible. Photogrammetry has been widely used for architectural preservation and also for creating in-game assets. Photogrammetric reconstruction to generate realistic images to enhance model training has also been explored in [15]. For geolocalization purposes, the authors in [16] proposed a scalable localization method using multimodal synthetic data. The advantage of using photogrammetric methods is that existing data and non-360 images of the area of interest can be leveraged. An example of such data is the Virtual Singapore project, which includes a textured 3D digital map of Singapore [17].

In this work we propose a workflow that combines the use of panoramic 360-degree images for image-based localization, and enhancing the training process using synthetic images generated from a photogrammetric 3D model. Our method can take advantage of the recent advances in 360-degree cameras which boasts real-time image stabilization and stitching of images, and output a standard equirectangular image (2:1 aspect ratio). Unlike the 6-DOF relocalization used in the PoseNet papers, we only focus on the 3D XYZ cartesian position for this work due to the nature of the panoramic image representation.

II. Data Collection and Preparation

In this section, we present the workflow used for obtaining the training data, for both real aerial images and the generated synthetic data.

A. UAV-mounted Camera

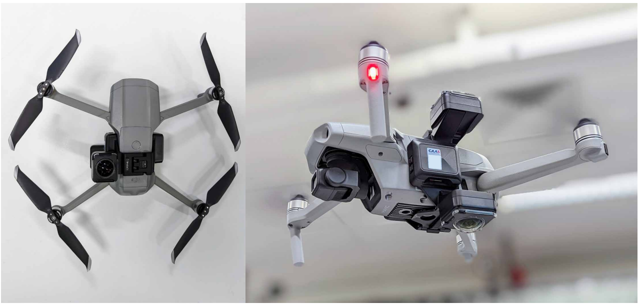


Fig. 2. DJI Air 2 with INSTA360 Sphere Camera. Top View (Left) and Side View while flying (Right).

To obtain the required 360-degree images, we opt to use a DJI AIR 2 together with an INSTA360 Sphere camera. The benefit of the INSTA360 Sphere system is that it provides an unobstructed 360-degree view, as the drone is sandwiched between the two lenses, essentially hiding it from view. The advantage of this setup is the ability to simultaneously capture normal non-360 images to enhance the 3D reconstruction for the synthetic data generation.

To obtain the image-position correspondence, the flight log was downloaded and the GPS location of the UAV is extracted. As the images are time-stamped, we can then match them with the corresponding flight data. The data used in this work was collected over two days, both around late afternoon and was relatively overcast. To represent the full 360-degree image in a 2D space, we would use the popular equirectangular projection throughout this work.

B. 3D Model Reconstruction

As normal non-360 images are also captured via the DJI drone camera, we can reconstruct a 3D model of the scene via photogrammetry techniques. A total of 223 12-Megapixel geotagged images were used for the reconstruction.

RealityCapture was the software used for the reconstruction. We used a ground control point in the middle of the scene to act as the reference for the rest of this work. The GPS coordinates are also converted to XYZ-coordinates with the reference point being in the middle of the environment.

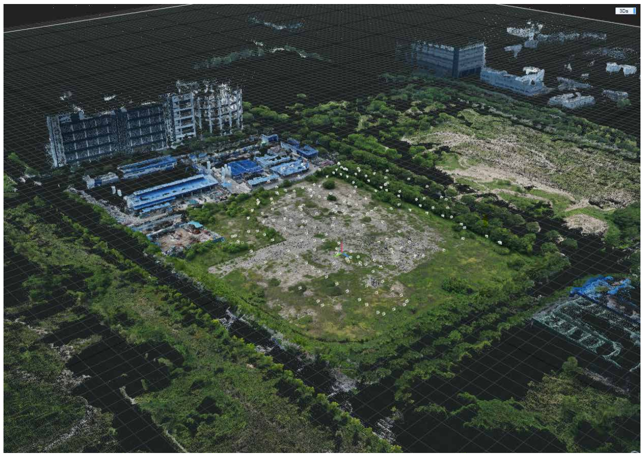


Fig. 3. Image Alignment and 3D Reconstruction.

C. Generating Sample Points and Synthetic Data

To generate the synthetic data, we position virtual 360 cameras in the reconstructed 3D scene and render the images. In our workflow, the 3D model is imported into a Digital Content Creation (DCC) software, in our case we chose Houdini by SideFX for its procedural workflow. The positions of the virtual 360 cameras are randomized following a uniform distribution within a specified bounding box, which we can define as the operating area of the UAV. The number of images to generate is also set beforehand.

A panoramic sky background is also added in the virtual scene to make the data more realistic. A randomized rotation of the skies is also applied with each synthetic image.

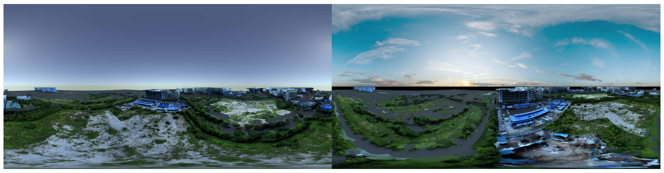


Fig. 4. Examples of the rendered synthetic data with different skies in equirectangular format.

III. Localization Network and Training

A. Localization Network

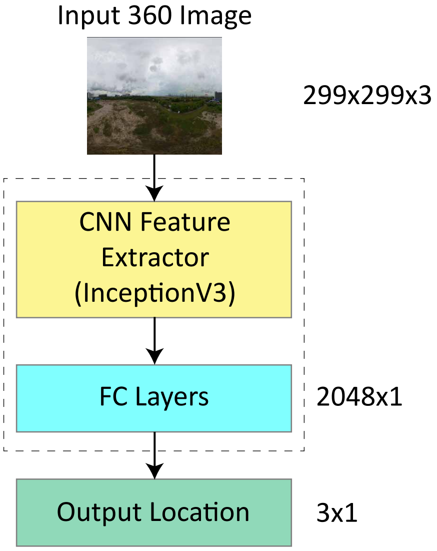


Fig. 5. Localization Network Flow.

The network consists of a main image feature extractor, which condenses the input image into a 2048-size vector, and feeds into 2 fully connected layers for the position regression, similar to the PoseNet architecture mentioned in the introduction. InceptionV3 [18] pretrained on ImageNet was used for the CNN, with a global max pooling layer added.

B. Training

We use Mean Square Error (MSE) as the loss function for training:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{1}$$

The NAdam optimizer [19] was used with a low learning rate of 0.0001 and $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-}8$. The images in the training datasets are also randomly shuffled between each epoch, and are fed into the training as minibatches of 24 images. Weights in the CNN feature extractor were also enabled for fine-tuning.

C. Data Augmentation

Several image augmentation techniques are employed to prevent the network from overfitting given the limited number of images. We used the Albumentations [20] library to implement the augmentations into our workflow. The parameters of the augmentation techniques are determined heuristically. Standard rotation and affine transform augmentations which are usually applied for classification and segmentation networks were not used here.

1) Hue Saturation Value: The hue, saturation and value of the input image are randomly varied to help the network generalize better.

2) Equirectangular Image Translation: Transforming and wrapping an equirectangular image horizontally is equal to rotating the 360 image sphere about the vertical axis. In this way we try to get the network to not overfit to a certain orientation.

3) Random Rain: Random Rain applies slight blurring and random raindrops to add noise to the image, essentially attempting to simulate different environmental conditions without doing so physically. This augmentation technique has been used in autonomous vehicle scenarios. [21].

4) Coarse Dropout (Blackout): This augmentation technique randomly 'blocks' part of the image by setting the area to black, as described in [22]. This can be thought of as simulating a partial obstruction of the 360 image.

Examples of the augmentations listed above are individually presented in Fig. 6.

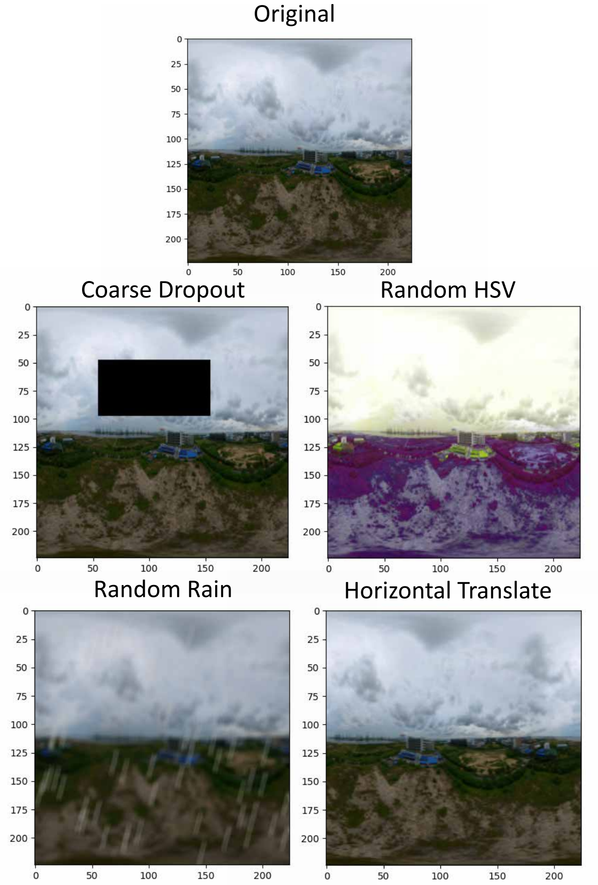


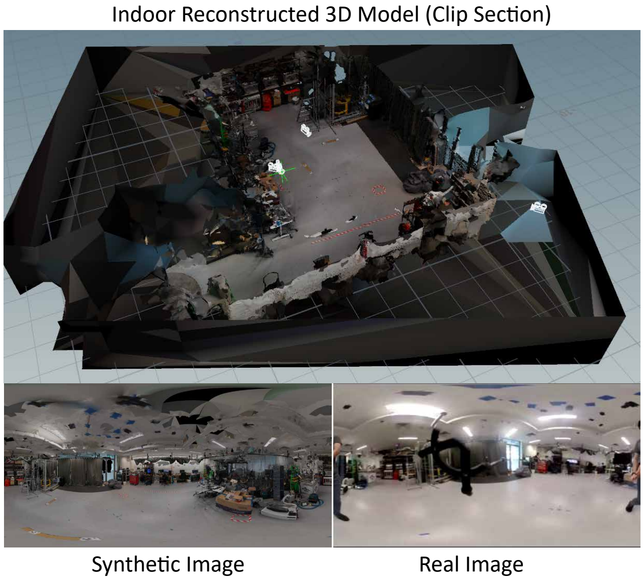
Fig. 6. Image Augmentation Examples.

IV. Experiments

We conducted two main experiments using our workflow. The first is in an indoor lab environment, and the second is an outdoor field.

A. Results (Indoor)

For this work, we started with a small proof-of-concept in a controlled indoor environment with an abundance of visual features. For this case, we went with full synthetic data for the training (around 400 images). The indoor lab reconstruction, and examples of the respective synthetic and real test images used are presented in Fig. 7.

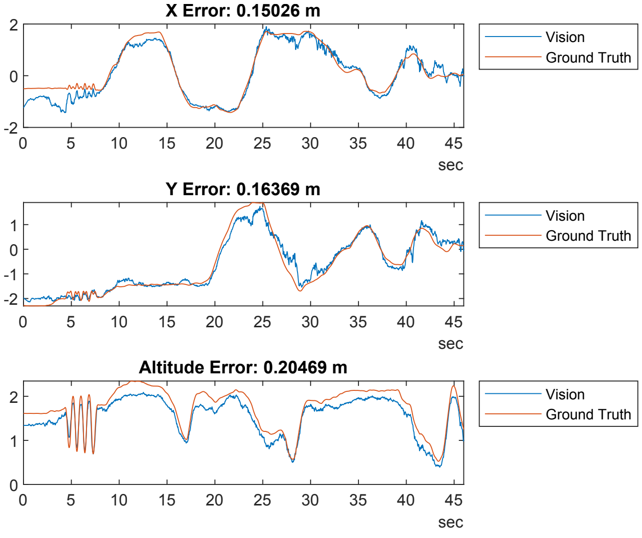

Indoor Reconstructed 3D Model (Clip Section)

Synthetic Image

Real Image

Fig. 7. Top: The reconstructed 3D model of the indoor lab environment. Left: An example of a synthetic image. Right: The real images used for testing.

The real 360-image data for the indoor environment was captured with an INSTA360 One X mounted on a selfie stick with motion capture markers mounted on it to obtain the ground truth. The coordinate systems of the 3D reconstuction and the motion capture system are synchronized using ground control points.


X Error: 0.15026 m

Y Error: 0.16369 m

Altitude Error: 0.20469 m

Fig. 8. Plots of the vision localization data against the ground truth motion capture data for all three axes. (Results are from the training set with data augmentation.)

Similar image augmentation was applied for the indoor dataset except for the RandomRain function. From the results presented in Fig. 8, we can observe that predicted values track the ground truth quite closely. There is also a large offset for the first 5 seconds in the X- and Z-axis. We suspect this is due to the obstruction caused by the human operator and part of the motion capture marker mount. A 3D plot of the testing space and the trajectories are presented in Fig. 9. As displayed in Table I, augmenting the dataset reduces the error by at least half.

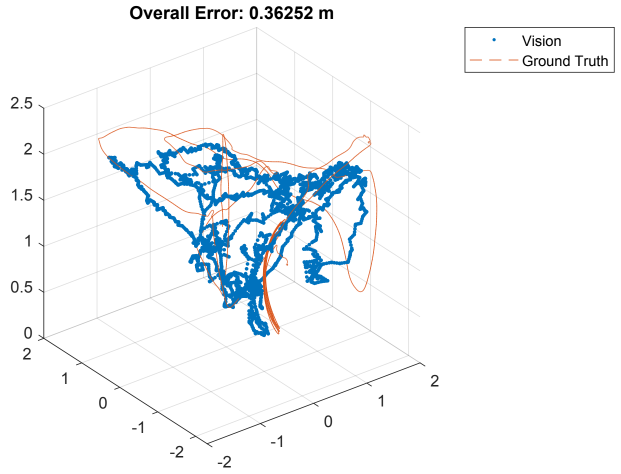

Overall Error: 0.36252 m

Fig. 9. 3D Plot of the indoor test of the vision data against the ground truth. This result is from the configuration with data augmentation.

	Med. X err.	Med. Y err.	Med. Z err.	Med. distance err.
No augmentation	0.3648	0.4880	0.1744	**0.7063**
With augmentation	0.1503	0.1639	0.2047	**0.3625**

TABLE I
MEDIAN ERRORS FOR THE INDOOR TESTS.

B. *Results (Outdoor)*

For the outdoor part, we chose an open space in the southern part of Singapore (Tuas South Avenue 16) as the testbed for our workflow. The environment is mostly empty field with a sparse collection of low-lying buildings and spans an area of around 250 m by 220 m (5.5 Ha). It can be noted that this area is rather featureless and thus can pose quite a challenge to image-based methods.

The respective outdoor dataset is separated into Train and Validation (80-20) split. An additional independent test set is also prepared and only contains real images. This will be used as the benchmark for comparing the performance of the different training configurations.

C. *Performance Comparisons and Discussion*

We quantitatively compare the results between the different training configurations, as displayed in Table II. Across the three different combinations of real and synthetic data, adding the image augmentations help to reduce the median error by at least 8 to 15 percent. The configuration with the mixed synthetic and real data outperformed configurations with only

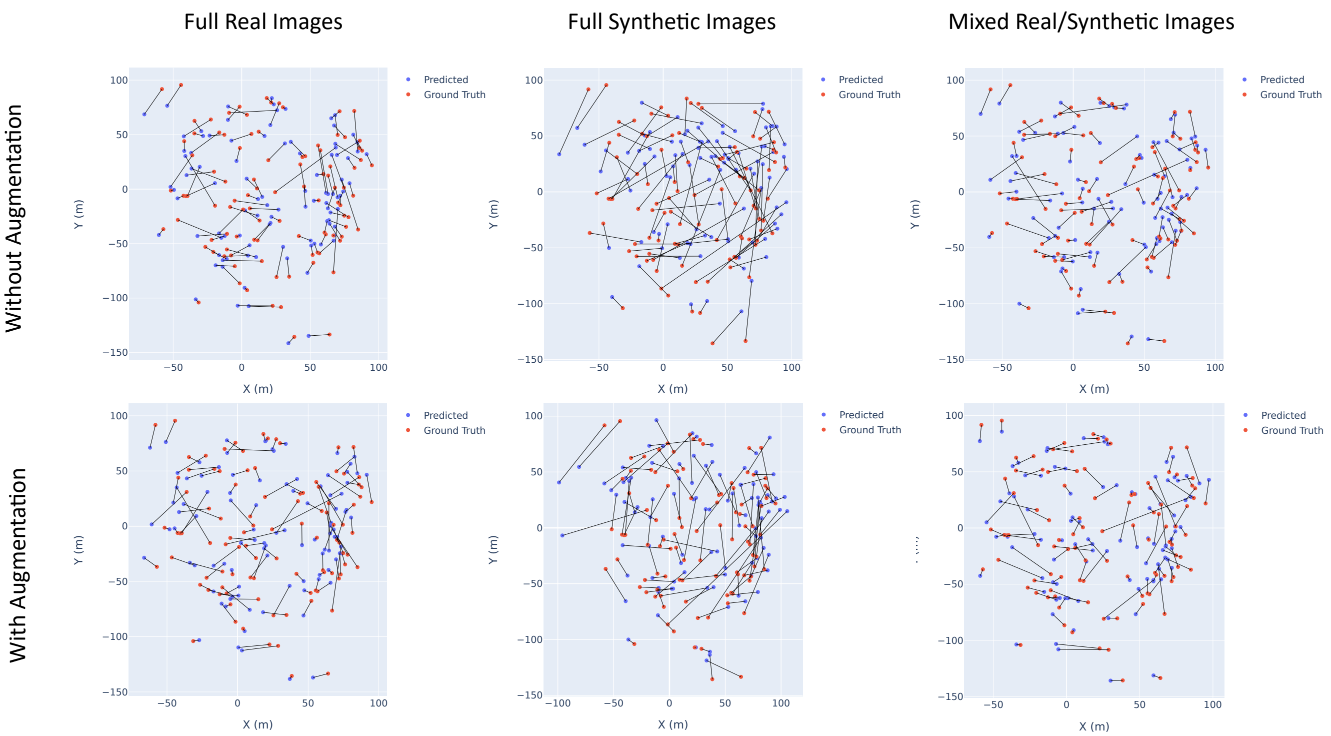


Fig. 10. XY Plot (equivalant to a top view) of each of the different training configurations on the test dataset. The test dataset is uniformly sampled from the initial images to cover the entire volume of interest. The top row are the results from training without augmentation, and the bottom row are from the configurations with augmentation.

Train Config.	X-Median Err. (m)	Y-Median Err. (m)	Z (Altitude)-Median Err. (m)	Median Distance Err. (m)
Full Real	7.616	9.250	2.784	16.334
Full Synthetic	19.526	22.617	9.522	39.209
Mixed 50%	**7.025**	**9.112**	1.583	15.235
Augmented Real	8.846	9.411	2.398	17.724
Augmented Synthetic	11.852	20.539	6.488	32.556
Augmented Mixed 50%	**7.102**	**7.787**	2.167	13.641

TABLE II
COMPARISON OF THE MEDIAN ERROR REPORTED FOR EACH CONFIGURATION.

real or synthetic data, and the ones with only synthetic data performing much worse. We suspect that the reconstruction alone might not possess enough detailed features for the network to localize on, but it is able to help the network generalize better when mixed with the real images. The seemingly high error might also stem from the location being rather featureless with little to no buildings surrounding the area, compared to the indoor environment discussed in the previous section, which possess features from top to bottom. We also suspect that while the real data contain more detailed images, mixing synthetic data helps improve performance as the network is further exposed to image location correspondences that are not present in the real image dataset.

Focusing on the XY position as presented in Fig. 10, we can qualitatively observe that the configuration with image augmentation and mixed data performs better, with the predictions being closer to the ground truth point. Also, we can notice that the error lines generally increase on the right side of the graph, which corresponds to the location with lots of similar-looking trees away from the main building on the left side, and we suspect that the lack of color difference might contribute to the larger error on the right side.

When looking at the top ten worst images across all the training configurations, (visualized in Fig. 11 with the corresponding ground truth location), we can see a few trends which we suspect might cause the localization to fail. The first is the distortion in the images, the more obvious ones being test image number 79 and 13. This distortion comes the built-in stabilization of the 360-camera that we use, most probably from the high acceleration that the drone experiences, which might cause an issue with the stabilization. As our workflow relies on the built-in stabilization, these stabilization failures would cause an issue with the localization performance, and thus would have to be resolved if it were to be implemented in real-time.

Other images also include the lack of ground features which

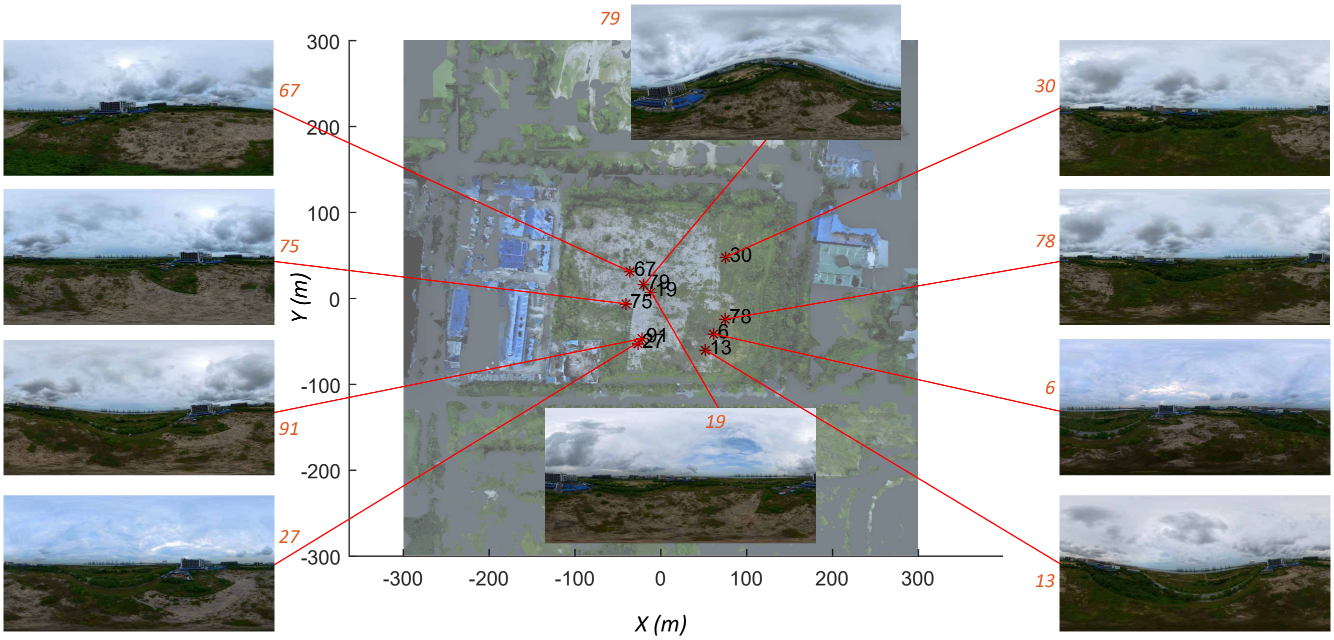


Fig. 11. Ten images with the highest error across all training configurations. Notice the large image distortions and the lack of features in some images.

might confuse the localization network. For example in test images 30 and 78, most of the image is just grass. Also, the resolution of 299x299 being fed into the network might not be enough to discern the small buildings in the background. As 360-degree images encompass the entire surrounding view, more data are packed into the same resolution compared to an image with a narrower field of view. A possible way to mitigate this is to crop out the sky and stretch out the bottom half of the image to give more resolution to the more important features on the ground. The possibility of exploring other 360-degree projections can also be considered, for example, using a stereographic instead of an equirectangular projection.

We also ran the localization network on a sample continuous UAV flight representative of a typical operation in the area. The results are presented in Fig. 12, with a top view XY plot as well as the Z-altitude plot. It can be observed that the offset (error) is more pronounced on the extreme left of the plot, which might be due to the limitations mentioned earlier.

V. Conclusion

We have demonstrated the viability of using 360-degree panoramic images for end-to-end aerial visual localization, and the possibility to utilize 3D reconstruction data to enhance the localization performance, in both indoor and outdoor environments, achieving a median distance error of 0.3625 m and 13.641 m respectively. In the future, we plan to explore different panoramic image representations, and also deploy it on an actual drone to perform real-time inference and closed-loop control. We also plan to experiment with different loss functions/feature extractors during training, and evaluate their performance. Additionally, different network architectures can also be explored.

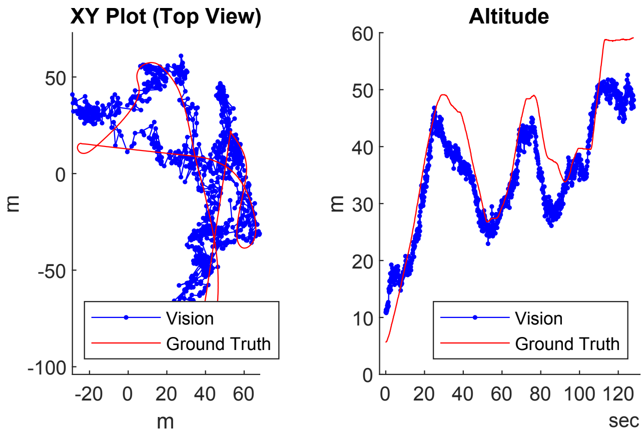


Fig. 12. XY Plot (equivalant to a top view) and Altitude (Z) of a sample continous UAV flight, representative of a typical UAV flight through the area. (Using Mixed and Augmented data configuration).

Acknowledgment

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-100E-2021-083) and by A*STAR under its Supply Chain 4.0 - Digital Supply Chain Development via Platform Technologies (Award M21J6a0080). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and A*STAR.

References

[1] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2D-to-3D matching," in *2011 International Conference on Computer Vision*. IEEE, 11 2011, pp. 667–674.

[2] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 12 2015, pp. 2938–2946. [Online]. Available: http://ieeexplore.ieee.org/document/7410693/

[3] A. Kendall and R. Cipolla, "Geometric Loss Functions for Camera Pose Regression with Deep Learning," 4 2017. [Online]. Available: http://arxiv.org/abs/1704.00390

[4] R. Zhang, Z. Luo, S. Dhanjal, C. Schmotzer, and S. Hasija, "Posenet++: A CNN Framework for Online Pose Regression and Robot Re-Localization," Tech. Rep. [Online]. Available: https://posenet-mobile-robot.github.io/

[5] C. Cimarelli, D. Cazzato, M. A. Olivares-Mendez, and H. Voos, "Faster Visual-Based Localization with Mobile-PoseNet," Tech. Rep.

[6] W. Harvey, C. Rainwater, and J. Cothren, "Direct aerial visual geolocalization using deep neural networks," *Remote Sensing*, vol. 13, no. 19, 10 2021.

[7] A. Shetty and G. X. Gao, "UAV Pose Estimation using Cross-view Geolocalization with Satellite Imagery," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2019, pp. 1827–1833. [Online]. Available: https://ieeexplore.ieee.org/document/8794228/

[8] T. YASHIRO, H. HIRAYAMA, and K. SAKAMURA, "An Indoor Localization Service using 360 Degree Spherical Camera," in *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*. IEEE, 3 2020, pp. 17–18. [Online]. Available: https://ieeexplore.ieee.org/document/9081315/

[9] A. R. Zamir and M. Shah, "Accurate Image Localization Based on Google Maps Street View," 2010, pp. 255–268.

[10] V. N. Murali and J. M. Coughlan, "Smartphone-based crosswalk detection and localization for visually impaired pedestrians," in *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, 7 2013, pp. 1–7.

[11] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras, "OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas," Tech. Rep. [Online]. Available: http://vcl.iti.gr/360-dataset/

[12] P. Barmpoutis, T. Stathaki, K. Dimitropoulos, and N. Grammalidis, "Early Fire Detection Based on Aerial 360-Degree Sensors, Deep Convolution Neural Networks and Exploitation of Fire Dynamic Textures," *Remote Sensing*, vol. 12, no. 19, p. 3177, 9 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/19/3177

[13] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for Data: Ground Truth from Computer Games," 8 2016.

[14] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 961–972, 9 2018.

[15] R. Lopez-Campos and J. Martinez-Carranza, "ESPADA: Extended synthetic and photogrammetric aerial-image dataset," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7981–7988, 10 2021.

[16] Q. Yan, J. Zheng, S. Reding, S. Li, and I. Doytchinov, "CrossLoc: Scalable Aerial Localization Assisted by Multimodal Synthetic Data," 12 2021. [Online]. Available: http://arxiv.org/abs/2112.09081

[17] "Virtual Singapore - National Research Foundation." [Online]. Available: https://www.nrf.gov.sg/programmes/virtual-singapore

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 12 2015.

[19] T. Dozat, "INCORPORATING NESTEROV MOMENTUM INTO ADAM," Tech. Rep., 2016.

[20] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," *Information*, vol. 11, no. 2, p. 125, 2 2020.

[21] G. Volk, S. Muller, A. v. Bernuth, D. Hospach, and O. Bringmann, "Towards Robust CNN-based Object Detection through Augmentation with Synthetic Rain Variations," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 10 2019, pp. 285–292.

[22] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random Erasing Data Augmentation," 8 2017. [Online]. Available: http://arxiv.org/abs/1708.04896