2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

Direct Aerial Visual Localization using Panoramic Synthetic Images and Domain Adaptation

Danial Sufiyan, Luke Soe Thura Win, Shane Kyi Hla Win, U-Xuan Tan, and Shaohui Foong

***Abstract*— In the realm of aerial vehicle navigation, the reliance on satellite-based and external localization methods presents vulnerabilities to various interferences. This drives the necessity for a self-sufficient absolute navigational system. Image-based localization methods, particularly Absolute Visual Localization (AVL), directly determine the pose in the global frame from a given image. A workflow using 360-degree panoramic images for image-based localization, driven by a Deep Convolutional Neural Network (DCNN), is proposed. Utilizing panoramic imagery offers the advantage of encompassing visual information from all angles. Synthetic data generated from multiple sources such as photogrammetry, Open Street Map (OSM), and official 3D building data are used to train the localization network. Domain adaptation using cycleGAN is also used to bridge the Sim2Real gap and enhance model performance. Utilizing OSM features are shown to improve localization performance (median Euclidean error) by at least 13%, and a further 20% with cycleGAN dataset augmentation. Closed loop control is also achieved using a trained model, enabling a quadrotor prototype to hover within a 1 m circle.**

I. Introduction

With the increasing popularity of aerial vehicles, whether manned or unmanned, significant research has been devoted to developing a more robust navigational suite. These vehicles primarily rely on a combination of an Inertial Measurement Unit (IMU) and the Global Navigation Satellite System (GNSS) to establish their location in three-dimensional space. However, there is a growing need for a self-sufficient absolute navigational system, given the vulnerability of satellite-based and external localization methods to a range of interferences, including natural obstacles such as weather and multi-pathing, as well as artificial disruptions such as jamming and spoofing.

Image-based localization methods can be divided into Relative Visual Localization (RVL), where the pose difference between two successive frames is estimated, and Absolute Visual Localization (AVL), where the pose in the global frame is directly determined. In this work, we specifically focus on the AVL aspect, aiming to directly localize the absolute pose from a given image.

One of the earliest works involving AVL include [1], [2] and [3], which directly regress 6DOF pose from a single image, attempting to solve the ”kidnapped robot” problem. Using 360 panoramic images for this is rather attractive as visual data from all around can be used to infer the location. In [4], the authors use 360-images to implement an indoor localization service. Other works using 360-images for navigation include [5], [6].

As using deep learning-based methods usually require a rather dense distribution of training data, synthetic data can provide a controlled and versatile alternative to purely real-world datasets. In the realm of AVL, one such work [7] utilizes synthetic data generated from a 3D model to train a localization network in an indoor environment.

However, depending on the type and quality of the synthetic data, there is a simulation-to-reality (Sim2Real) gap which needs to be bridged for the trained network to be deployed in the actual environment. This is where generative AI for domain adaptation/style transfer such as Pix2Pix [8] for paired images and CycleGAN [9] for unpaired images can be leveraged. The work in [10] successfully utilizes a Generative Adversarial Network (GAN) for domain adaptation between unpaired simulated and real images. [11] instead utilizes paired 360° images to transform a multi-channel image from a 3D scene into a photorealistic street-view image.

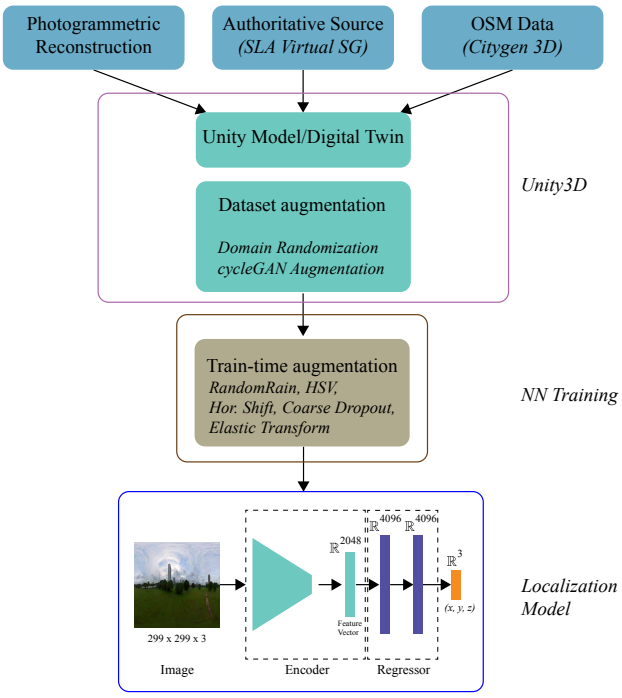

Fig. 1. Summary of the workflow presented in this work. Multiple data sources can be used to build 3D model in unity to generate the required synthetic panoramic images. Multiple augmentations are applied to enable the network to generalize better to real data. (SLA Virtual SG = Singapore Land Authority (SLA) Virtual Singapore [12]).

The authors are with the Engineering Product Development Pillar, Singapore University of Technology & Design (SUTD), 8 Somapah Road, Singapore 487372. (Corresponding e-mail: foongshaohui@sutd.edu.sg)

This can be paired with generating visual features/buildings based on OpenStreetMap (OSM) data. OSM is a publicly available map of the world, containing crowd-sourced data of streets, buildings, and other features. The authors in [13] were one of the first to integrate OSM data directly for autonomous robot navigation. Works in [14] and [15] leverage OSM data to localize a vehicle equipped with a 3D-lidar scanner to navigate without prior mapping.

Our previous work [16] introduced the possibility of using panoramic images for localization, and focused on using real panoramic images for training. In this work, we focus on expanding the possibility of using only synthetic panoramic images for training, utilizing multiple data sources including non-panoramic images. A visual summary of the workflow is presented in Fig. 1. Also, our previous work only contained a single outdoor environment, thus we expand our workflow to additional outdoor environments. We also demonstrate the capability for the trained network to be deployed on an actual drone.

A. Contributions

- Demonstrate building a panoramic visual localization pipeline using data from multiple sources
- Training a localization model using only synthetic panoramic images
- Implemented domain adaptation for data augmentation
- Evaluation of trained model at multiple locations
- Demonstrate closed-loop control via deployment on an Unmanned Aerial Vehicle (UAV)

II. Data Generation

In this work, we rely on several data sources to build the synthetic data generation pipeline. Using these data sources, we can build a visually accurate 3D model that can be used to generate the images. The advantage of this method is that the data can come in many forms and is not restricted to be in the form of a panoramic image.

A. Data Sources

1) Self-Collected Photogrammetry/Image Data: Using a camera-equipped drone, both normal and panoramic images can be acquired, utilizing the normal images for photogrammetric reconstruction to build a 3D model where more synthetic panoramic image data can be generated. Fig. 2 shows the drone setup used for data collection. To establish a correspondence between the taken image and the location ground truth, an RTK GPS logger was mounted on the drone.

To build the photogrammetry 3D models, RealityCapture [17] software was used. Presented in Fig. 3 are overviews of the 3D models built for the different environments.

2) Authoritative Sources: With the increasing push toward smart cities, authorities such as the Singapore Land Authority (SLA) possess 'digital twin' data of the city such as high fidelity 3D models and textures of the building facades.

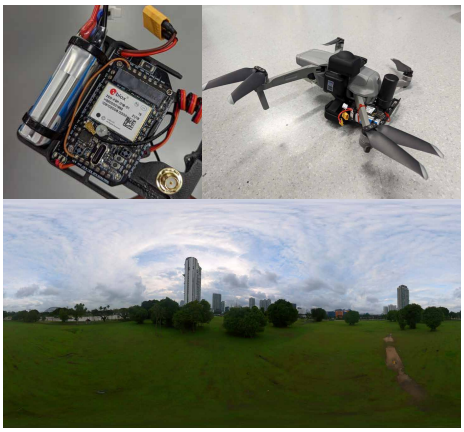

Fig. 2. *Top Left*: RTK GPS logger used for ground truth collection. *Top Right*: Drone with mounted Insta360 Sphere camera and RTK GPS logger. *Bottom*: Example of collected panoramic 360 image in 2:1 equirectangular format (from Kallang Riverside).

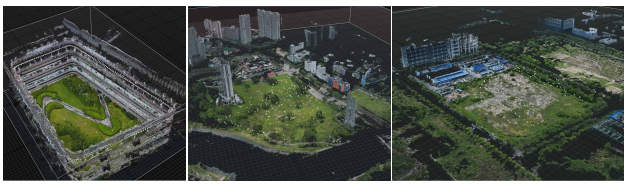

Fig. 3. Photogrammetric model for the different environments. Left: AerialArena@SUTD, Middle: Kallang Riverside, Right: Tuas South.

3) Generation using OpenStreetMap Data: We opted to utilize CityGen3D [18], a plugin within Unity that converts OpenStreetMap (OSM) data into a 3D landscape. Normally used for procedural game world generation based on real life locations, we propose that it can also be utilized to fill in incomplete visual data. Features such as roads can be procedurally generated and combined with the photogrammetric or building facade data, either self-collected or from authoritative sources. A visual example of this is presented in Fig. 4.

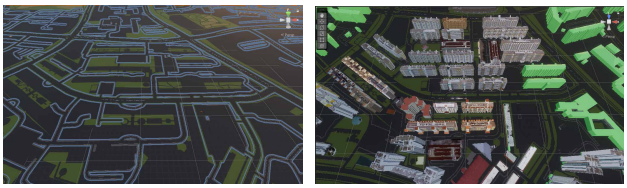

Fig. 4. *Left*: Generated roads by CityGen3D using OSM data. *Right*: CityGen3D + SLA Building Model and Facade Data. Buildings in green are generated by CityGen3D using OSM Building information.

B. Rendering of Visual Images

For our synthetic data generation pipeline, we opted to use Unity, taking advantage of its real-time rendering. The Unity Perception [19] module was used for domain randomization of the environment, such as the skybox, sun angle, lighting, etc.. This domain randomization is built into the synthetic dataset. Data points also randomized following a uniform distribution within a specified bounding box. Data points that

collide with buildings are removed from the datasets. Some examples of the generated images are shown in Fig. 5.

Fig. 5. Example synthetic images generated in Unity (Kallang Riverside), presented in 2:1 equirectangular format. Environmental conditions such as lighting, skies and sun angle are randomized.

C. Environments

In this work, several environments in different areas of Singapore were selected for testing. Each of them possesses a unique set of features that serves as a testbed for our AVL model.

1) Aerial Arena @ SUTD: The Aerial Arena is an enclosed and netted semi-outdoor testing area located within the university compound. Approximately 1000 m^2.

2) Kallang Riverside Park: A small open field located near the Singapore National Stadium, bounded by two condominium complexes. Approximately 47,000 m^2.

3) Tuas South Ave 16: An industrial estate with low-lying buildings and a large field with relatively low amount of features. Approximately 51,000 m^2.

4) 3D Virtual Singapore Dataset - Ang Mo Kio (AMK): Courtesy of the Singapore Land Authority (SLA) [12], a high fidelity 3D model of Singapore was obtained, encompassing the Ang Mo Kio region, covering approximately a 1 km x 1 km area. Using this 3D model, synthetic panoramic image data can be generated. Due to confidentiality reasons, only the building facade texture data was provided, with no terrain and contour data. In this work we focus on a smaller area of about 150,000 m^2, along Ang Mo Kio St 31.

D. Dataset Format

The ground truth location for each environment is projected to a flat XYZ format in metres from the lat-lon-altitude (LLA) format of each image. Each environment has its own unique reference point to be used as the origin of the projection. A summary of the datasets used is presented in Table I.

III. Localization Method

A. Network Architecture

The network consists of a feature extractor network (Xception) followed by a fully connected regression network. A dropout layer of 0.5 is added before each dense layer (dimension of 4096). The final output layer is of dimension 3. ReLU activation is used for the dense layers and linear activation for the output. This is visualized in Fig. 6.

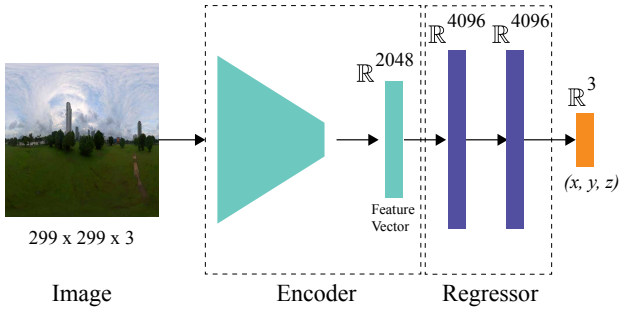

Fig. 6. Localization network architecture - Input image is fed to an encoder to be represented as a feature vector, where the fully connected regressors output the position estimate.

B. Training

The framework for building and training the model is Keras (Tensorflow). A batch size of 48 and learning rate of 0.000075 is used.

1) Loss Function: Mean absolute error (MAE) is used as the loss function for training.

2) Train-time Data Augmentation: A number of image augmentation techniques during train-time were employed to make the network generalize better. We use RandomRain, HSV, Horizontal Shift (with Wraparound), Coarse Dropout, and Elastic Transform as described in [16]. The Albumentations library [20] was used for this portion.

3) Auto-Augment: AutoAugment [21] is a technique employed during train time to automatically learn certain image augmentation. We utilize AutoAugment transfer which uses learned policies from the *Reduced-CIFAR10* dataset instead of learning from scratch. The operations included in the AutoAugment policy are: *[Color, Equalize, ShearY, Brightness, Sharpness, AutoContrast, Rotate, Posterize, Contrast, Invert, Solarize]*. These are mostly color-based transformations, and for our purpose, translation augmentation is removed.

C. Data Augmentation using CycleGAN

Fig. 7. Comparison between original synthetic image and the same image put through cycleGAN. Images are re-scaled from 2:1 equirectangular format to 1:1 aspect ratio to fit the network requirements.

Compared to the other 3 environments where photogrammetric reconstruction was used, the AMK dataset provided by SLA is visually quite different compared to the actual location. In order to bridge the this gap, we introduce a style transfer data augmentation technique leveraging Generative Adversarial Networks (GANs). cycleGAN [9] utilizes

TABLE I. Summary of all datasets. Test images are all real images.

Environment	Area (m^2)	#Images (train + val)	split	#Images (Test)	Training Data Sources	Origin Point (LLA)
AerialArena@SUTD	1,000	7,500	85/15	51	Photogrammetry	$[1.341007°, 103.962742°, 68.394]$
Kallang Riverside	47,000	7,500	85/15	201	Photogrammetry	$[1.298913°, 103.783326°, 19.840]$
Tuas South	51,000	7,500	85/15	154	Photogrammetry	$[1.266157°, 103.625434°, 27.594]$
Ang Mo Kio (St 31)	150,000	13,422	85/15	177	Authoritative, OSM Generated	$[1.360934°, 103.843622°, 8.297]$

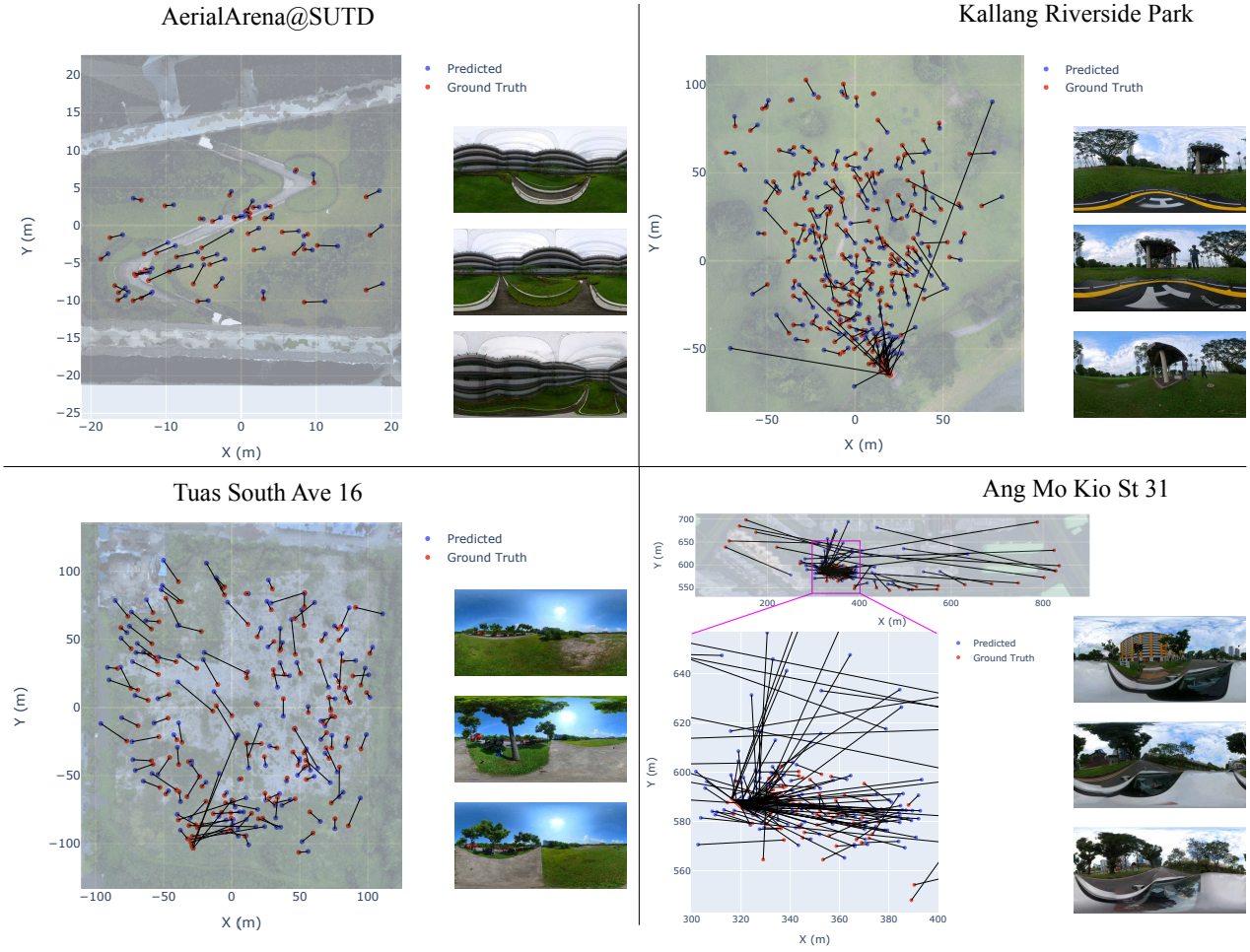

Fig. 8. XY Plot (Top View) for each environment on their respective test sets. The lines connect the predicted point to the respective ground truth. Displayed on the right of each plot are the images with the top 3 highest error.

unpaired images to synthesize a style transfer relationship between the original and target images and vice versa. We utilize this to enrich dataset diversity by capturing visual features and color palettes not present in the original synthetic dataset. Furthermore, the training images do not need to have a ground truth label for them to be used. Training data for this was collected via a combination of aerial and ground videos. We used Keras to train and deploy the cycleGAN network. Training was run for 300 epochs and a learning rate of 0.000075 was used. The output resolution was set to 384×384 pixels. Fig. 7 illustrates an example of an image before and after being put through the trained cycleGAN network.

IV. Results and Discussion

A. Performance on Independent Test Set

This section presents the results of the different configurations. An independent test set is reserved for each environment and contains only real images. Images in this test set were not used during the training process for both the localization and cycleGAN networks. Median Euclidean Distance error is used as the performance metric for comparison. It is calculated as: $Median\left(\sqrt{(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2+(z_i-\hat{z}_i)^2}\right)$. The localization performance of each environment is presented in Table II.

From the images with the highest error presented in Fig. 8, we can see that most of the images are near the ground level. We suspect that this is due to limited visibility

TABLE II. Summary of the median error reported for each configuration.

Environment	Med. Euclid. Err (m)	X-Med. Err. (m)	Y-Med. Err.	Z (Altitude)-Med. Err.
AerialArena	1.496	1.234	0.503	0.537
Kallang Riverside	8.874	3.772	4.692	3.920
Tuas	12.142	6.619	6.504	2.445
Ang Mo Kio	18.616	13.717	4.763	3.055

of the surrounding environment, and also features that are present at that proximity to the ground might not match the features that the trained network is looking for. We also present a histogram of the euclidean errors with the ground truth altitude in Fig. 9. As seen from the histograms, the localization performance for most of the environments drastically decreases at lower altitudes near the ground. For most of the outdoor environments, the localization starts to be more stable at around 10-15 m above ground, where more of the surrounding features start to be visible and consist most of the image, enabling the network to localize better. We suspect this discrepancy manifests less in the smaller AerialArena@SUTD environment as the surrounding features are much closer to the inference points.

For the AMK test set, the aerial images consist of only a smaller area due to flight permit restrictions. We supplement this with additional ground data to span the area trained for the localization. The aerial data is concentrated between 300-400 m (x-coordinates) and 520-620 m (y-coordinates), which is depicted by the zoomed in view shown in Fig. 8. As seen in the Figure, the localization failure happens when too many of the important features are obstructed, in this case by trees and other roadside objects.

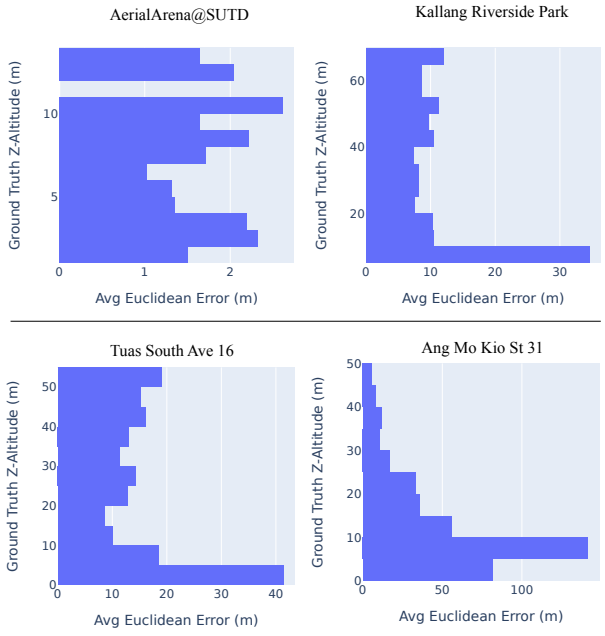

Fig. 9. Histogram breakdown of the euclidean errors with the altitude (ground truth).

B. Experiments on AMK environment

Several variants of the Ang Mo Kio dataset were created, with different combinations of the OSM generated entities to study their effect on localization performance. Each dataset variant was trained with the same amount of iterations. In the basic variant, only the data provided by SLA is included, and the ground color is added into the domain randomization process. The second variant includes only OSM generated roads. The third includes roads and generated buildings with the default textures included by CityGen3D. The fourth variant randomizes the color of the building facade, and the last configuration contains the synthetic images put through the cycleGAN domain adaptation step, and combined with the un-augmented image. The OSM generated buildings are used to fill up the spaces where there are no SLA data provided. Sample images from each variant is presented in Fig. 10, where the additional generated buildings missing from the SLA dataset can be seen.

As seen from the results in Table III, adding additional generated features reduced the localization error, compared to the pure AMK dataset with only the building facade. Using the median Euclidean error as a metric, adding the generated OSM roads helped to improve the error by around 13%, and including the OSM generated buildings with random facade color improved it by a further 5%. Interestingly, using the default texturing scheme built into the CityGen3D program made the performance worse. This might be due to the network learning the wrong textural features on the building facade instead of the building outlines. When put through the cycleGAN domain adaptation process, the performance greatly improved by a further 20% compared to the dataset with randomized building colors. The biggest improvement difference occurs between the pure building-only dataset and the one with the generated roads. We can infer that in a dense urban area like this, the network not only looks for visual features on the building facades, but also the road layouts to determine its position.

C. Closed-Loop Control

In this section, we test the capability of the trained localization network to provide closed-loop positional reference to an actual UAV. The UAV was outfitted with a Ricoh Theta X 360 camera, which features onboard stabilization and outputs panoramic images in equirectangular format, and is fed to an on-board computer for processing. The predicted location is then fed to a quadrotor running the PX4 Firmware [22]. The onboard computer communicates the predicted position to the flight controller via MavLink. The setup is summarized in Fig. 11. The block diagram is presented in Fig. 12.

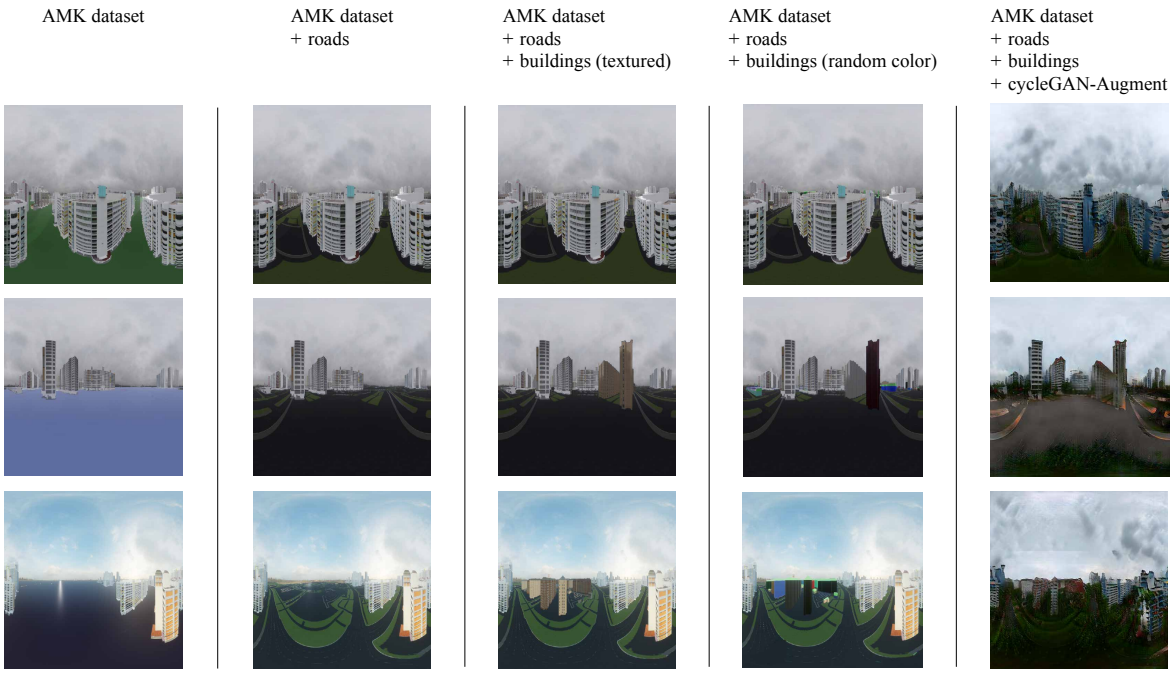

Fig. 10. Presented here are image examples of the different variants of the AMK dataset. Images are re-scaled to 1:1 ratio for standardization. Generated road textures can be observed in the 2nd column. Missing buildings not present in the raw SLA dataset are generated via CityGen3D and OSM data (3rd column onwards).

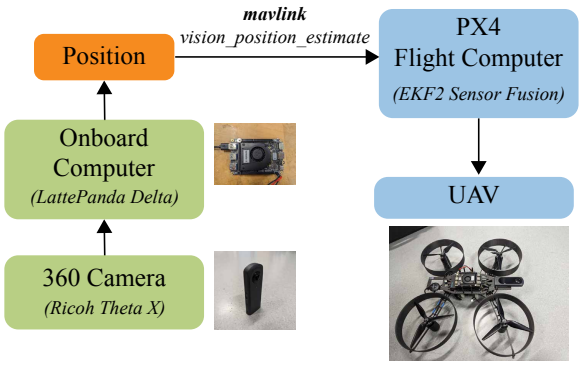

Fig. 11. Setup diagram for closed-loop control using the onboard 360 camera.

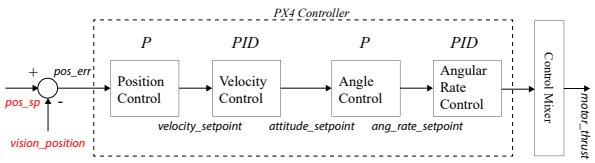

Fig. 12. Block diagram for closed-loop control.

We demonstrate a simple hover utilizing the onboard 360 camera for localization. The onboard network inference time is ∼700ms. Figure 13 shows the setpoint together with the filtered position data estimated by the PX4 EKF2 filter (extended Kalman filter). Fig. 14 shows the raw unfiltered position as output from the localization network. Onboard GPS is disabled. As seen from Fig. 13, the UAV manages to successfully hover about a single point using the visual localization network as the only absolute positional reference system. Together with the EKF, the UAV manages to stay within a 1 meter circle. A step change in the Y-desired position is input at around the 800 s mark. As seen in the graph, the UAV successfully adjusts its position to the new setpoint. This test was done in the AerialArena@SUTD environment.

V. Conclusion

In this work, we successfully demonstrate the feasibility of utilizing synthetic data generated from multiple data sources in creating a neural network AVL model. The workflow is successfully tested and validated on multiple environments. Different levels of generated features using OSM data and the effect on localization performance was also studied.

TABLE III. Results for the different variants of the AMK dataset.

	Median Euclid. Err, (m)
Pure AMK dataset	28.985
+ roads	25.255
+ roads + buildings (textured)	42.306
+ roads + buildings (random color)	23.850
+ roads + buildings (textured) + images w/ cycleGAN-Augment	18.616

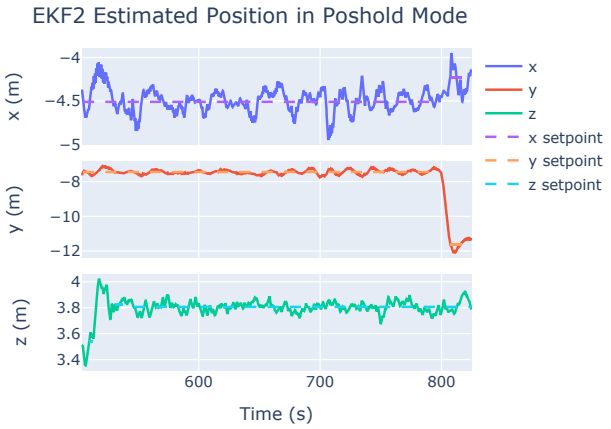

Fig. 13. XYZ positions estimated by PX4 EKF2 using visual localization data and their respective setpoints.

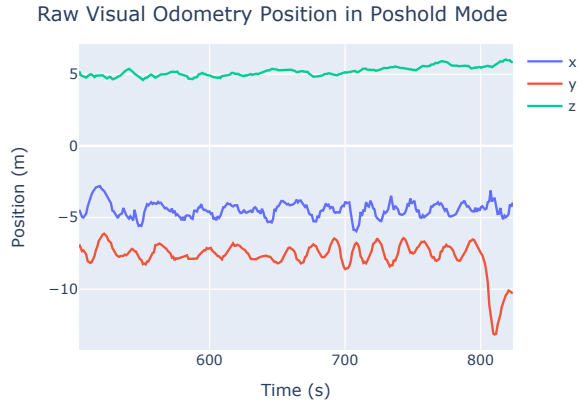

Fig. 14. Raw XYZ position from the localization network, inferred on the image stream from the onboard 360 camera

Model performance was shown to be enhanced by at least 13% (using Median Euclidean Error) via domain adaptation (cycleGAN). Closed-loop control on a quadrotor was also demonstrated using a trained AVL model. Further research directions can include testing on night lighting conditions, experiment with different panoramic representations, and explore other network architectures.

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] 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/

[2] 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/

[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] 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.

[5] 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.

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

[7] D. Acharya, K. Khoshelham, and S. Winter, "BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 245–258, 4 2019.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 11 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[9] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," 3 2017. [Online]. Available: http://arxiv.org/abs/1703.10593

[10] D. Acharya, C. J. Tatli, and K. Khoshelham, "Synthetic-real image domain adaptation for indoor camera pose regression using a 3D model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 202, pp. 405–421, 8 2023.

[11] M. Bresson, Y. Xing, and W. Guo, "Sim2Real: Generative AI to Enhance Photorealism through Domain Transfer with GAN and Seven-Chanel-360°-Paired-Images Dataset," *Sensors*, vol. 24, no. 1, p. 94, 12 2023.

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

[13] M. Hentschel and B. Wagner, "Autonomous robot navigation based on OpenStreetMap geodata," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 9 2010, pp. 1645–1650. [Online]. Available: http://ieeexplore.ieee.org/document/5625092/

[14] Y. Cho, G. Kim, S. Lee, and J.-H. Ryu, "OpenStreetMap-Based LiDAR Global Localization in Urban Environment Without a Prior LiDAR Map," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4999–5006, 4 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9716862/

[15] F. Yan, O. Vysotska, and C. Stachniss, "Global Localization on OpenStreetMap Using 4-bit Semantic Descriptors," in *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 9 2019, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/8870918/

[16] D. Sufiyan, Y. H. Pheh, L. Soe Thura Win, S. K. Hla Win, U. X. Tan, and S. Foong, "Panoramic Image-Based Aerial Localization using Synthetic Data via Photogrammetric Reconstruction," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, vol. 2023-June. Institute of Electrical and Electronics Engineers Inc., 2023, pp. 656–662.

[17] Capturing Reality, "Reality Capture." [Online]. Available: https://www.capturingreality.com

[18] CityGen Technologies Ltd, "CityGen3D." [Online]. Available: https://www.citygen3d.com

[19] S. Borkman, A. Crespi, S. Dhakad, S. Ganguly, J. Hogins, Y.-C. Jhang, M. Kamalzadeh, B. Li, S. Leal, P. Parisi, C. Romero, W. Smith, A. Thaman, S. Warren, and N. Yadav, "Unity Perception: Generate Synthetic Data for Computer Vision," 7 2021. [Online]. Available: http://arxiv.org/abs/2107.04259

[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] E. D. Cubuk, B. Zoph, V. Vasudevan, and Q. V. Le Google Brain, "AutoAugment: Learning Augmentation Strategies from Data," Tech. Rep., 2019. [Online]. Available: https://pillow.readthedocs.io/en/5.1.x/

[22] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2015, pp. 6235–6240.