2020 IEEE 16th International Conference on Control & Automation (ICCA)
Oct 9-11, 2020 (Virtual) Sapporo, Hokkaido, Japan

A Deep Learning Approach to Localization for Navigation on a Miniature Autonomous Blimp

Landan Seguin* Justin Zheng* Alberto Li Qiuyang Tao Fumin Zhang

Abstract— **The Georgia Tech Miniature Autonomous Blimp (GT-MAB) needs localization algorithms to navigate to waypoints in an indoor environment without leveraging an external motion capture system. Indoor aerial robots often require a motion capture system for localization or employ simultaneous localization and mapping (SLAM) algorithms for navigation. The proposed strategy for GT-MAB localization can be accomplished using lightweight sensors on a weight-constrained platform like the GT-MAB. We train an end-to-end convolutional neural network (CNN) that predicts the horizontal position and heading of the GT-MAB using video collected by an onboard monocular RGB camera. On the other hand, the height of the GT-MAB is estimated from measurements through a time-of-flight (ToF) single-beam laser sensor. The monocular camera and the single-beam laser sensor are sufficient for the localization algorithm to localize the GT-MAB in real time, achieving the averaged 3D positioning errors to be less than 20 cm, and the averaged heading errors to be less than 3 degrees. With the accuracy of our proposed localization method, we are able to use simple proportional-integral-derivative controllers to control the GT-MAB for waypoint navigation. Experimental results on the waypoint following are provided, which demonstrates the use of a CNN as the primary localization method for estimating the pose of an indoor robot that successfully enables navigation to specified waypoints.**

I. INTRODUCTION

GPS sensors cannot accurately locate flying robots operating in indoor environments, so external motion capture systems are frequently used for indoor navigation instead [1][2][3][4]. Motion capture systems provide fast and accurate pose estimation, but they require a direct line of sight to the robot. Another solution to indoor localization of aerial robots involves simultaneous localization and mapping (SLAM) [5][6][7][8][9]. As discussed in [10], SLAM suffers from the need to robustly track features, which can be especially difficult in dynamic environments or environments with scarce features. SLAM pipelines also often require the fusion of multiple sensors in order to produce a good pose estimate, which adds to the complexity of the implementation.

With recent advances in convolutional neural networks (CNNs) for camera pose estimation, we can exploit the learned feature extractors of CNNs in order to localize a robot within indoor environments without the need for motion capture systems or complex filtering algorithms. A CNN can learn to produce a pose estimate both indoors and outdoors, within environments as small as a few meters to environments as large as several thousand meters [11], whereas motion capture systems are typically used for localization in a constrained indoor environment. CNN-based localization methods can be efficient, while requiring fewer sensors.

Fig. 1: The Georgia Tech Miniature Autonomous Blimp (GT-MAB). The GT-MAB consists of two major components, the envelope and the gondola. The envelope is the upper balloon-like portion that is filled with helium so that the platform can fly for long durations. The gondola houses the sensors, motors, microcontroller, camera and communication modules.

In this paper, we develop CNN-based localization for the Georgia Tech Miniature Autonomous Blimp (GT-MAB) [12] (Fig. 1). The GT-MAB consists of an envelope and a gondola that hosts a microcontroller, four motors, two single-beam lasers, a wireless monocular camera, and potentially other devices. Previous experiments with the GT-MAB relied on a motion capture system in order to navigate to waypoints [12]. By using a CNN and laser sensor for localization, we show that such an aerial robot can navigate autonomously indoors without the use of a motion capture system or SLAM algorithms.

Our contribution is as follows. First, we add a time-of-flight (ToF) single-beam laser sensor to the underside of the GT-MAB, which allows the GT-MAB to estimate its height. Second, we utilize a CNN based on VGG-16 [13] to regress the GT-MAB's horizontal position and heading in an indoor environment. Finally, we provide experimental results, showing that an indoor aerial robot can navigate autonomously to waypoints using a CNN as the primary localization method, without the use of external localization systems or complex mapping and filtering algorithms. To the best of our knowledge, this work is the first use of a CNN as a primary localization method for an autonomous indoor aerial robot.

The remainder of this paper is organized as follows. Section II reviews relevant work including state of the art methods for CNN-based localization and visual odometry,

* Indicates equal contribution, names are in alphabetical order.

Landan Seguin, Justin Zheng, Alberto Li, Qiuyang Tao, and Fumin Zhang are with the School of Electrical and Computer Engineering, Georgia Institute of Techonology, Atlanta, GA 30308, USA. Email:{ljseguin, justin_zheng, albertoli, qiuyang, fumin}@gatech.edu

together with research previously conducted on the GT-MAB. Section III describes the related hardware of the GT-MAB which facilitates control, perception, and communication on the GT-MAB. Section IV describes the CNN used for localizing the GT-MAB in an indoor environment, along with the training and evaluation procedure. Section V presents the proportional-integral-derivative (PID) controllers designed for the GT-MAB so that it is able to navigate to waypoints. In Section VI, we provide experimental results of the GT-MAB navigating autonomously to complete a waypoint mission in an indoor room without external localization methods. Finally, we conclude our work in Section VII and discuss future research paths and applications.

II. RELATED WORK

Over the last several years, CNNs have shown a remarkable ability to accomplish tasks related to scene understanding and navigation. CNNs have previously been used for the task of place recognition [14][15]. Given a camera image, these CNNs predict a high-level location of the camera pose, which could be used as an input to a separate algorithm that produces a more refined pose estimate. CNNs have also recently been used to tackle the visual odometry (VO) problem. In [16], a recurrent convolutional neural network (RCNN) is used to learn monocular VO. The works [17][18] use CNNs to estimate depth maps and also tackle monocular VO in an unsupervised manner. While these works compare well against traditional geometric approaches to VO, the output pose drifts over time when loop closure techniques are not applied. In our application, we aim to directly predict an absolute pose using a CNN that does not drift over time, without aid from any external localization sources.

In [11][19][20], a CNN is used to directly output a 6 degree of freedom (6DoF) camera pose given a monocular RGB camera image. These works show that CNN-based localization can be fast (200 Hz) and produce accurate poses for both small indoor environments and large outdoor environments. Reference [11] also showed that a CNN takes advantage of large, textureless regions of an image to assist with pose estimation, whereas traditional methods like SLAM often require the detection and tracking of rich features. Visual structure from motion (VSFM) [21] is used to automate the data labeling process, showing that data can be easily collected and automatically labeled for an arbitrary environment. In the work [22], the authors use an RCNN to directly predict the position of a robot, although this is done in a simulation environment.

In all of the previously described works, the poses are not used to control or navigate a physical platform. In this paper, we use a localization method most similar to [11] to estimate the pose of the GT-MAB, which allows us to control the GT-MAB to a designated location indoors.

The GT-MAB has previously been used as a research platform for controls and Human-Robot Interaction (HRI) experiments. Our previous work [12] introduces the GT-MAB platform and discusses its dynamics and compares the platform to other indoor aerial robots. A main advantage of the GT-MAB and similar blimps [23][24][25][26][27] is that compared to other aerial vehicles, they are safer to humans and fly longer durations. In [28] and [29], the authors add a lightweight wireless monocular camera to the GT-MAB so that it is able to follow humans and recognize gestures, without the use of an external localization system. However, the localization in these works is done with respect to a designated human, and not based on pose estimation. We use the GT-MAB as the platform for our experiments because we are interested in exploring the feasibility and performance of using a CNN as the primary localization method for an aerial robot, which can potentially enable platforms like the GT-MAB to navigate in more indoor environments without the need for external localization methods or SLAM algorithms.

III. HARDWARE

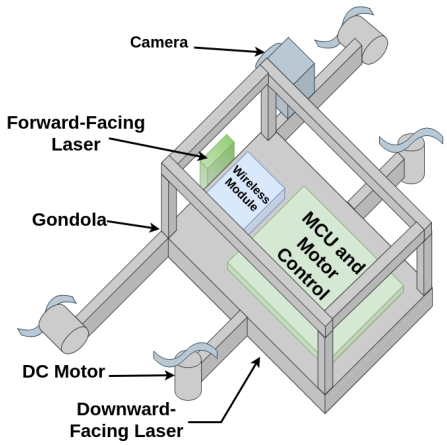


Fig. 2: GT-MAB gondola that houses motors, sensors, a microcontroller unit (MCU) to interface with the motors and sensors, and a wireless module used to send sensor data to the offboard base station desktop and also receive motor commands.

Designed as in [12][28][29], the GT-MAB has two major sections, the envelope and the gondola. The envelope has a saucer-like shape and is inflated with helium. This envelope is naturally cushioned, which makes this platform safe for human-robot interaction (HRI) applications. The gondola is shown in Figure 2. The housing carries the microcontroller, power supply, sensors, and motors. The gondola has two vertical motors for controlling height, and two forward-facing motors for rotation or forward-backward motion. The GT-MAB is supported by a Linux base station desktop through wireless communication, which has a graphics processing unit (GPU) for fast deep learning inference.

Like [28] and [29], we mount a wireless RGB camera on the front of the gondola to allow the GT-MAB to visually perceive the environment. The camera wirelessly transmits camera images captured on the GT-MAB to a receiver that is connected to the base station computer at a frame rate of 30 Hz. These camera images serve as inputs to our CNN for localization.

We add a single-beam laser to the underside of the gondola. For our experiments described later in this paper, the downward-facing laser is used to estimate the height of the GT-MAB, assuming there are no objects on the ground. This

laser can provide range measurements with errors averaging less than 1 cm. The power consumption, size, weight, and ease of use of the laser sensor make it very suitable for platforms like the GT-MAB.

IV. CONVOLUTIONAL NEURAL NETWORK FOR LOCALIZATION

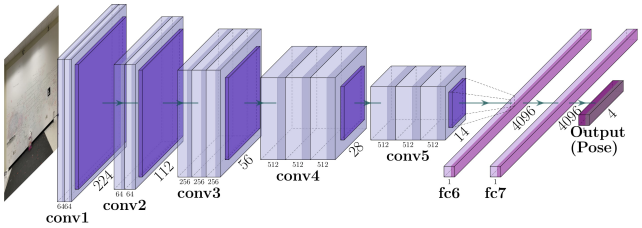


Fig. 3: CNN based on VGG-16 used to regress the position and heading of the GT-MAB in an indoor environment. The input to the CNN is a single 224x224 monocular RGB camera image, and the output is a 4-dimensional vector, consisting of x and y position, along with a unit vector to represent heading.

A. Architecture

In this paper, we design and train a CNN that takes a single RGB image as input and outputs a pose. We use VGG-16 as the backbone architecture for our localization CNN because VGG-16 has shown to capture spatial context very well. We remove the classification head of VGG-16 and replace it with a 4-element fully connected layer (Figure 3). The input to the network is a 224 x 224 RGB camera image, and the output is a vector v, which consists of the 2D horizontal position vector q and a heading θ which is represented as a unit vector.

$$v = [q, \theta] \tag{1}$$

Predicting a single heading value can cause difficulties during training. For example, if the network predicts a single heading value of 359° for a ground truth heading of 0°, an L1 loss would be large even though the error is actually quite small. As discussed in [20], a non-injective orientation representation can be challenging to learn, which is why the authors chose to use a quaternion to represent orientation. We do not predict roll and pitch because the GT-MAB typically does not operate outside of a narrow band of roll and pitch values due to its self-stabilizing design as described in [12]. Because we are only interested in predicting heading, we use the CNN to predict a unit vector, where the two components are continuous between $[-1, 1]$, and an L1 loss can be easily applied. The predicted vector can be normalized such that it produces a valid unit vector.

B. Data Collection

The environment in which we are experimenting with the GT-MAB is an indoor box-shaped room. This room has an Optitrack motion capture system [30], which locates unique markers that are placed on top of the GT-MAB in order to get a very accurate pose estimate of the GT-MAB. While methods like VSFM can be used as in [11] to collect ground truth poses, we can instead use the Optitrack system to provide ground truth poses without the required optimization and computation time of VSFM. VSFM could be used to automatically label data for future experiments outside of our laboratory environment.

Our data collection process requires minimal human effort. We manually move the blimp around the lab space while simultaneously capturing camera images and recording Optitrack ground truth poses. This means that our data is automatically labeled, where each camera image is paired with a corresponding ground truth pose. During data collection, we vary the GT-MAB's position and rotation angles so that we capture the expected operating space within the room. The Optitrack system is able to provide ground truth poses in an area that is a 3m x 3m square. Our data is collected over several periods that total to approximately 3 hours, where 150000 images are used for training and 14000 are used for testing. During the collection of this dataset, some objects in the room vary such as the absence or presence of equipment and people around the room. The GT-MAB operates in the center area of the room that is within the field of view of the Optitrack system, which is generally clear of any objects. One issue we face is that the onboard camera can sometimes produce glitchy images due to its wireless range, shown in Figure 4. However, we find that the network is generally robust to these glitches since many examples are present in the training data.

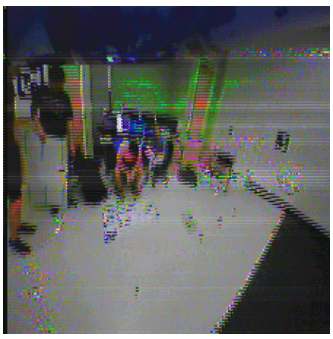


Fig. 4: Glitchy image captured by the monocular wireless RGB camera and received by the base station desktop. These glitches occur occasionally when there is poor communication signal between the camera and the receiver connected to the base station desktop.

C. Training

The network used in this paper is implemented in Pytorch [31]. We train the network for 25 epochs, with a batch size of 16. During training, we use the following L1 loss function,

$$Loss = \sum_{i}^{n} (||\hat{v} - v||_1) \tag{2}$$

L1 loss is used instead of L2 loss, allowing the network to converge faster. This is because our CNN's prediction errors quickly become less than 1, and an L2 loss would square these errors to produce diminishing loss values, whereas L1 loss would provide larger weight updates. The stochastic gradient descent (SGD) optimizer [32] is used during training, with a momentum of 0.9 and step interval of 1 epoch.

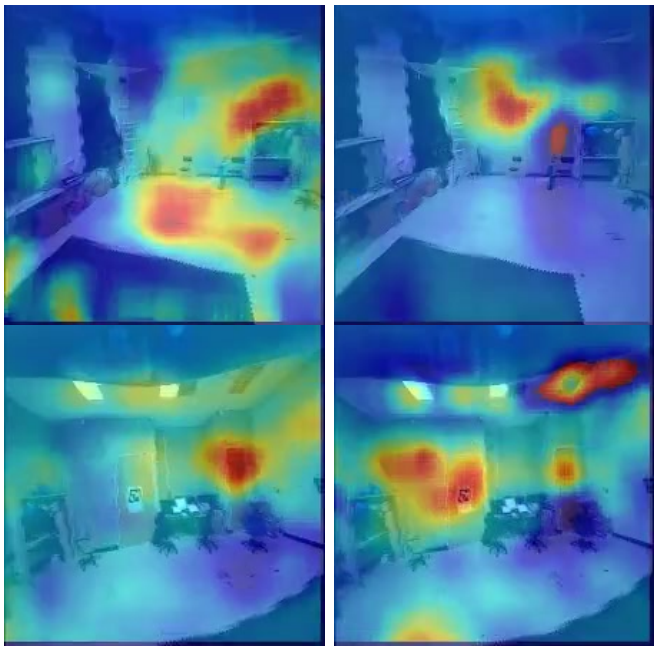
D. Network Performance

The network is able to run at 30 Hz, which is the max frame rate of the GT-MAB's camera. We evaluate our network on a test dataset consisting of 14000 images that were collected in separate trials from the training dataset. The results are shown in Table I.

TABLE I: Error in position and heading between the CNN predictions and the ground truth provided by Optitrack on the test dataset.

Error Values	Mean	Std. Dev
X position (m)	0.0447	0.0500
Y position (m)	0.0456	0.0522
2D position (m)	0.0732	0.0649
Heading (deg)	1.5298	3.7242

We are able to achieve centimeter-level position estimation error, and heading estimation error on the order of just a couple degrees. This error is significantly lower than the indoor performance results of [11] and [20], although this is likely because our testing environment is rather simple and less varying. Additionally, we are able to label our data using a precise motion capture system.



(a) (b)

Fig. 5: Input images to the CNN overlayed with a heatmap indicating regions that most influence the CNN prediction. (a) shows the heatmap for position error, and (b) shows the heatmap for heading error. "Hotter" regions indicate parts of the image that most affect the prediction error of the CNN.

E. Ablation Study

While we cannot fully understand the learned features of a CNN, we can perform an ablation study to gain some intuition. For this experiment, we slide a 10 pixel x 10 pixel black patch across test images and evaluate the change in error to the predicted position and heading that is caused by the patch. Using a heatmap, we can create a visualization of the pixels that result in the largest errors when zeroed out, shown in Figure 5.

We observe that some features of the room that are important to the CNN's prediction include smooth areas on the door, floor, and walls. This observation is consistent with the results of [11], which showed that large textureless regions of the scene can help improve localization estimates, whereas traditional geometric localization methods often fail to extract features in these regions. It is also important to note that objects that may have moved throughout data collection, such as the chairs or small objects on the floor, do not contribute significantly to the network's pose estimate as shown by the minor changes in prediction error when ablated.

We can see that ablated features affect position error and heading error differently, although there is some overlap in features.

V. CONTROLLERS

Our goal is to control the GT-MAB to a target waypoint in an indoor room. In order to accomplish this, we must consider the GT-MAB's dynamics. The GT-MAB's dynamics are described in detail in [12]. To control the GT-MAB to a waypoint, the following measurements and dynamics equations are required.

1. Distance:

$$\hat{d} = \sqrt{(\hat{x}_{cnn} - x_{wp})^2 + (\hat{y}_{cnn} - y_{wp})^2} \tag{3}$$

where $\hat{d}$ is the estimated distance between the GT-MAB and the target waypoint, $[\hat{x}_{cnn}, \hat{y}_{cnn}]$ are the predicted horizontal coordinates of the GT-MAB using the localization CNN, and $[x_{wp}, y_{wp}]$ are the horizontal coordinates of the target waypoint.

Assuming the GT-MAB is heading toward the waypoint, it can change its distance to the waypoint using its forward-facing motors as represented by the following model

$$m\ddot{d} = F_x + f_x \tag{4}$$

where d is the distance between the GT-MAB and the waypoint, F_x is the external force in the direction of the GT-MAB's forward-facing motors, and f_x is the force generated by the forward-facing motors.

2. Height:

$$\hat{h} = \hat{z}_{laser} - z_0 \tag{5}$$

where $\hat{h}$ is the estimated error between $\hat{z}_{laser}$, the predicted height of the GT-MAB from the laser sensor, and the target height z_0.

The GT-MAB can control its height using its vertical motors as represented by the following model

$$m\ddot{z} = F_z + f_z \tag{6}$$

where z is the height of the GT-MAB, F_z is the external force in the direction of the GT-MAB's vertical motors, and f_z is the force generated by the vertical motors.

3. Heading:

$$\theta_{wp} = \arctan(\frac{y_{wp} - \hat{y}_{cnn}}{x_{wp} - \hat{x}_{cnn}}) \tag{7a}$$

$$\hat{\psi} = \hat{\theta}_{cnn} - \theta_{wp} \tag{7b}$$

where θ_{wp} is the target heading angle for the GT-MAB, and $\hat{\psi}$ is the estimated error between the predicted heading angle from the localization CNN, $\hat{\theta}_{cnn}$, and the target heading angle θ_{wp}.

The GT-MAB is able to control its heading angle using its forward-facing motors for rotation as represented by the following model

$$I\ddot{\theta} = M + \tau \tag{8}$$

where θ is the GT-MAB's heading angle, M is the external moments exerted on the GT-MAB, and τ is the torque generated by the forward-facing motors.

We design 3 PID controllers similarly as [28] to control the GT-MAB. The distance controller uses $\hat{d}$ as feedback in order to produce the control signal f_x. The height controller uses $\hat{h}$ as feedback in order to produce the control signal f_z. The heading controller uses $\hat{\psi}$ as feedback in order to produce the control signal τ. For navigation to a waypoint, the goal for the controllers is to have $\hat{d} = 0$, $\hat{h} = 0$, and $\hat{\psi} = 0$.

We tune the PID parameters in MATLAB based on the system identification of the GT-MAB conducted in [12].

VI. WAYPOINT FOLLOWING

To present the capabilities of controlling the GT-MAB using a CNN and single-beam laser sensor for localization, we establish a waypoint following task. For this task, we set four waypoints in the same room where data was collected. The waypoints form a 2m x 2m square. The GT-MAB is controlled using the PID controllers described in Section V. The GT-MAB navigates to each of the four waypoints sequentially in the counter-clockwise direction. Once the GT-MAB is within 0.25m of the target waypoint, the GT-MAB will continue on to the next waypoint. We set the target height z_0 to 1.8 m.

The CNN described in this paper is used to output the horizontal position and heading of the GT-MAB, while the downward-facing laser sensor is used to obtain the height. We use the Robot Operating System (ROS) [33] in a Linux environment on the base station desktop to run core software. A microcontroller is used onboard the GT-MAB to read data from the laser sensor and control the motors, while a wireless module communicates information between the GT-MAB and base station desktop. Ground truth measurements are obtained through the vrpn_client_ros package [34] which reads position and heading estimates from the Optitrack motion capture system.

The predicted pose estimates of the GT-MAB during the waypoint following task are shown in Figures 6 and 7. The predicted heading is shown in Figure 8. Ground truth values are also provided.

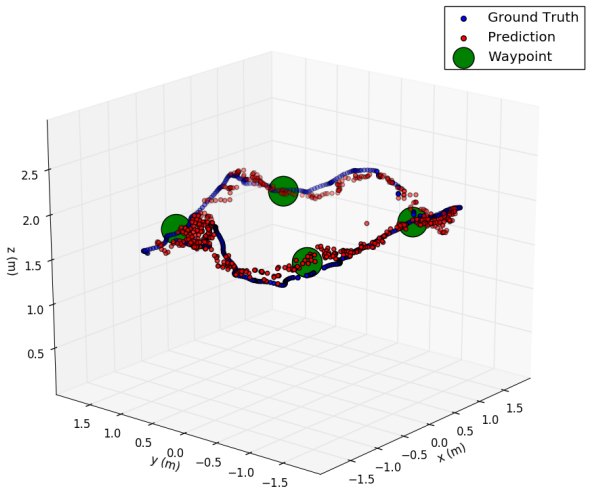


Fig. 6: Predicted and ground truth 3D position of the GT-MAB during the waypoint following task. The green spheres indicate each waypoint that the GT-MAB must navigate to before moving onto the next waypoint.

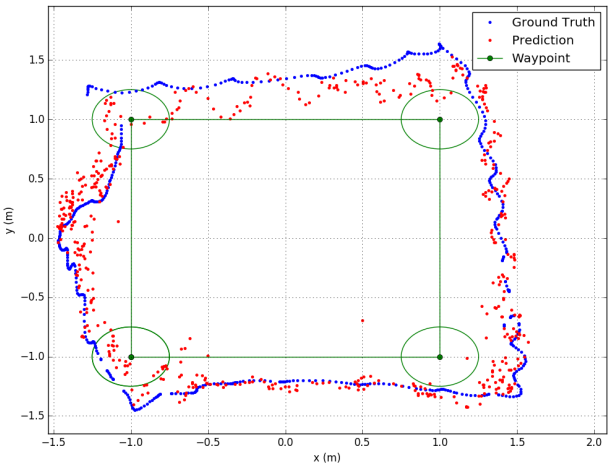


Fig. 7: Top-down view (2D) of predicted and ground truth position of GT-MAB during waypoint following task.

The GT-MAB takes 55 seconds to complete a single lap around the 4 waypoints. During the waypoint following task, our localization method is able to provide pose estimates of the GT-MAB at 30 Hz, which is the max camera frame rate. The pose estimation error values are shown in Table II. We are able to achieve average horizontal position errors of less than 18 cm using a single onboard camera sensor, whereas many SLAM algorithms require the fusion of multiple sensors like IMU, camera images, and laser sensors. Our localization method does not drift over time, and does not require a map. As shown in Section IV-E, the localization CNN is able to make use of textureless regions of the room, such as the floor and walls, along with more distinct features like ceiling lights and doors in order to produce an accurate horizontal position estimate. The downward-facing laser sensor is able to estimate the GT-MAB's height with average errors of less than 4 cm.

The localization error during the waypoint following mis-

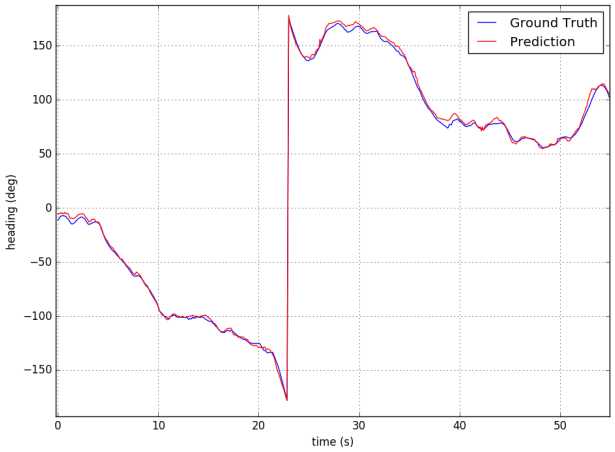


Fig. 8: Predicted and ground truth heading of GT-MAB during waypoint following mission. The CNN is able to correctly handle the angle wrap-around that occurs around $t = 23$ seconds.

TABLE II: Error mean and standard deviation of predicted position and heading of the GT-MAB during the waypoint following task.

Errors	μ	σ
X position (m)	0.0670	0.0615
Y position (m)	0.1499	0.1072
2D position (m)	0.1776	0.1035
Height (m)	0.0378	0.0249
3D position (m)	0.1976	0.1045
Heading (deg)	2.5471	2.9233

sion is higher than the test set error from Section IV. This is likely due to several reasons. First, the waypoint mission was completed several weeks after the training data was collected, so different aspects of the room may have changed. Additionally, the motion of the GT-MAB during the waypoint mission may have resulted in camera images outside the distribution of data collected for training. The error in predicted height from the laser sensor is higher than its specifications because of slight rolling and pitching during the GT-MABs flight, which changes the measured distance to the ground. While there is a performance difference in localization, we find that the errors are small enough such that the GT-MAB is still able to navigate to each of the waypoints.

In this experiment, we show that the GT-MAB is able to successfully navigate around an indoor room using a CNN to predict its horizontal position and heading, and a downward-facing laser sensor to predict its height.

VII. SUMMARY

We have presented a localization method for the GT-MAB that does not require an external motion capture system or complex SLAM and filtering algorithms for navigation. Instead, we describe a CNN that can be used to regress the horizontal position and heading of the GT-MAB, and the integration of a small, light-weight laser sensor that can accurately predict the height of the GT-MAB. We have shown that with this localization method and several PID controllers that control the GT-MAB's position, height, and heading, the GT-MAB is able to autonomously navigate to waypoints within an indoor environment. The waypoint following task shows that our localization method is accurate enough to use as an input to PID controllers in order to control the GT-MAB to different locations in a room. This work serves as a foundation for future experiments using the CNN localization output as an additional measurement for sensor fusion algorithms. We can also explore localizing and controlling the GT-MAB in larger indoor environments using our localization method, such that the GT-MAB could potentially guide people to different rooms within a building.

VIII. ACKNOWLEDGEMENT

The research work is supported by ONR grants N00014-19-1-2556, N00014-19-1-2266 and N00014-16-1-2667; NSF grants OCE-1559475, CNS-1828678, and S&AS-1849228; NRL grants N00173-17-1-G001 and N00173-19-P-1412 ; and NOAA grant NA16NOS0120028.

The authors would like to thank Devleena Das and Adam Kinsel, who helped conduct experiments.

REFERENCES

[1] A. Mashood, A. Dirir, M. Hussein, H. Noura, and F. Awwad, "Quadrotor object tracking using real-time motion sensing," in *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, Dec 2016, pp. 1–4.

[2] J. Zou, C. Wang, and Y. M. Wang, "The development of indoor positioning aerial robot based on motion capture system," in *2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, Nov 2016, pp. 380–383.

[3] J. P. How, B. Behihke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems Magazine*, vol. 28, no. 2, pp. 51–64, April 2008.

[4] S. Al Habsi, M. Shehada, M. Abdoon, A. Mashood, and H. Noura, "Integration of a vicon camera system for indoor flight of a parrot ar drone," in *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, Dec 2015, pp. 1–6.

[5] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," 10 2012, pp. 2815–2821.

[6] R. Huang, P. Tan, and B. M. Chen, "Monocular vision-based autonomous navigation system on a toy quadcopter in unknown environments," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2015, pp. 1260–1269.

[7] A. Bachrach, S. Prentice, R. He, and N. Roy, "Range–robust autonomous navigation in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 5, pp. 644–666, 2011.

[8] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, Feb 2012.

[9] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slambased navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20412

[10] Y. Wang, W. Zhang, and P. An, "A survey of simultaneous localization and mapping on unstructured lunar complex environment," *AIP Conference Proceedings*, vol. 1890, no. 1, p. 030010, 2017. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.5005198

[11] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[12] S. Cho, V. Mishra, Q. Tao, P. Vamell, M. King-Smith, A. Muni, W. Smallwood, and F. Zhang, "Autopilot design for a class of miniature autonomous blimps," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug 2017, pp. 841–846.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," *Proceedings of Robotics: Science and Systems XII*, 2015.

[15] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.
[16] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2043–2050.
[17] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7286–7291.
[18] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.
[19] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4762–4769.
[20] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[21] C. Wu, "Towards linear-time incremental structure from motion," in *2013 International Conference on 3D Vision - 3DV 2013*, June 2013, pp. 127–134.
[22] I. Suginaka, H. Iizuka, and M. Yamamoto, "Robustness of mobile robot localization using recurrent convolutional neural network," in *2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES)*, Nov 2017, pp. 95–100.
[23] J. Mller and W. Burgard, "Efficient probabilistic localization for autonomous indoor airships using sonar, air flow, and imu sensors," *Advanced Robotics*, vol. 27, no. 9, pp. 711–724, 2013. [Online]. Available: https://doi.org/10.1080/01691864.2013.779005
[24] P. González, W. Burgard, R. Sanz Domínguez, and J. López Fernández, "Developing a low-cost autonomous indoor blimp," 2009.
[25] A. Elfes, S. Siqueira Bueno, M. Bergerman, and J. G. Ramos, "A semi-autonomous robotic airship for environmental monitoring missions," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 4, May 1998, pp. 3449–3455 vol.4.
[26] M. Burri, L. Gasser, M. Kch, M. Krebs, S. Laube, A. Ledergerber, D. Meier, R. Michaud, L. Mosimann, L. Mri, C. Ruch, A. Schaffner, N. Vuilliomenet, J. Weichart, K. Rudin, S. Leutenegger, J. Alonso-Mora, R. Siegwart, and P. Beardsley, "Design and control of a spherical omnidirectional blimp," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1873–1879.
[27] D. St-Onge, C. Gosselin, and N. Reeves, "Dynamic modelling and control of a cubic flying blimp using external motion capture," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 229, no. 10, pp. 970–982, 2015. [Online]. Available: https://doi.org/10.1177/0959651815597603
[28] N. Yao, E. Anaya, Q. Tao, S. Cho, H. Zheng, and F. Zhang, "Monocular vision-based human following on miniature robotic blimp," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3244–3249.
[29] N.-s. Yao, Q.-y. Tao, W.-y. Liu, Z. Liu, Y. Tian, P.-y. Wang, T. Li, and F. Zhang, "Autonomous flying blimp interaction with human in an indoor space," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 1, pp. 45–59, Jan 2019. [Online]. Available: https://doi.org/10.1631/FITEE.1800587
[30] NaturalPoint, "Optitrack." [Online]. Available: optitrack.com
[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
[32] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.
[33] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.
[34] R. M. Taylor, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser, "Vrpn: a device-independent, network-transparent vr peripheral system," in *VRST*, 2001.