Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 643–650

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

Deep Neural Network for Autonomous UAV Navigation in Indoor Corridor Environments

Ram Prasad Padhy*, Sachin Verma, Shahzad Ahmad, Suman Kumar Choudhury, Pankaj Kumar Sa

Department of Computer Science and Engineering, National Institute of Technology, Rourkela 769008, India

Abstract

In recent years, the UAV technology is unceasingly emerging as a revolutionary reform among the research community. In this paper, we propose a method that facilitates UAVs with a monocular camera to navigate autonomously in previously unknown and GPS-denied indoor corridor arenas. The proposed system uses a state-of-the-art Convolutional Neural Network (CNN) model to achieve the task. We propose a novel approach, which uses the video feed extracted from the front camera of the UAV and passes it through a deep neural network model to decide on the next course of maneuver. The entire process is treated as a classification task where the deep neural network model is responsible for classifying the image as left, right or center of the corridor. The training is performed over a dataset of images, collected from various indoor corridor environments. Apart from utilizing the front facing camera, the model is not dependent on any other sensor. We demonstrate the efficacy of the proposed system in real-time indoor corridor scenarios.

Keywords: UAV, Monocular, GPS, Convolution, Deep Neural Network, Classification Task

1. Introduction

Unmanned Aerial Vehicles (UAV) have gained huge popularity among the research community for their practical applications in many fields such as aerial surveillance, precision agriculture, military, intelligent transportation, search and rescue operations, and many more. Profound progress has been made to make the navigation of these devices autonomous. Global Positioning System(GPS) [1] is actively serving for outdoor autonomous navigation. However, GPS is either inactive or not so prominent in indoor scenarios and hence, brings out many challenges for autonomous navigation [22].

In recent years, many solutions for autonomous navigation in indoor scenarios have been put forward. Most of these solutions rely upon laser range finders (LIDARs) [26], RGB-D sensors [13], stereo vision [17] to create a 3D

* Corresponding author. Tel.: +91-9899190334.
E-mail address: ramprasad.nitr@gmail.com

1877-0509
10.1016/j.procs.2018.07.099

Fig. 1. A quadcopter UAV navigating autonomously in an indoor corridor environment

map of an unknown environment, which in turn helps in finding the relative position of the device at any instant of time. This process is popularly known as Simultaneous Localization and Mapping (SLAM) [9, 11, 31] in literature. However, this is a computationally heavy process, and hence fails in most of the real-time scenarios. In this paper, we describe a system that enables a quadcopter to travel autonomously in indoor surroundings without much computational overhead. Our system does not require any external sensory aid or equipment other than an inbuilt monocular camera [2, 6]. Our approach is to train a classifier using deep learning techniques that impersonates an expert pilot action, to choose flight commands for autonomous navigation of UAVs in indoor corridor scenarios.

Figure 1 shows a moving UAV in a corridor environment. The proposed model will use a popular Convolutional Neural Network (CNN) [16] architecture, DenseNet-161 [14], that will train our custom dataset. The classifier will receive input from the quadcopter's front camera and in turn, will return a navigation command to the quadcopter in order to take right decision for autonomous navigation. The salient feature of our approach relies on the fact that it will help the quadcopter to navigate autonomously without the use of other computationally heavy methods like SLAM or heavy-weight sensors like LIDARs. This approach will help the quadcopter to navigate in extreme scenarios where 3D map generation of the surrounding can be erroneous.

The contributions of the paper can be enumerated as below:
(1) We propose a deep learning architecture for autonomous indoor navigation in corridor scenarios.
(2) We provide our custom dataset. The dataset contains images at different positions from various lengths of the corridors.

The remaining part of this paper is described in following sections. Section 2 presents an overview of the prior research done in this field. Section 3 explains about the custom dataset creation and the CNN architecture used to accomplish the task. Section 4 demonstrates our experiments in real world corridor scenarios. Finally, section 5 offers concluding remarks.

2. Related Work

Previous works on autonomous UAV navigation have mainly focused on the creation of a 3D map of the surrounding environment. Some methods deal with accurate quadcopter control [19, 20] in known environments. These methods, however, learn an advanced [9, 11, 31] tracking scheme, and hence restricting their use in a lab environments only. Other approaches learn map from a previously transcribed manual flight and the quadcopters are made to fly the same trajectory [21] again. Most of the outdoor flights use GPS-based pose estimation, which is not so reliable in indoor scenarios.

Some techniques use range sensors like infrared sensors, or RGB-D sensors [13], or laser rangefinders. Roberts *et al.* [27] used a single ultrasonic sensor with an infrared sensor to propose an autonomous navigation method which was even capable of collision avoidance. Bry *et al.* [7] proposed a state estimation method with a LIDAR and inertial measurement unit (IMU) to navigate autonomously in GPS denied unknown environments. The problem with range

sensors is that they are heavy-weight sensors and also, the power requirement is very high. Hence, these are not suitable for most of the light-weight aerial vehicles.

SLAM technique uses range or visual sensors to design a 3D map [9, 11, 31] of the unknown environment in transit of the flights, along with finding the device location in the map [18] at any instant. Bachrach *et al.* [4] utilized a laser rangefinder to implement SLAM for generating the 3D map of an unknown indoor scenario. Celik *et al.* [8] showed indoor navigation using monocular camera [2, 6] with the help of SLAM technique. The major drawback with SLAM is that the 3D map regeneration of the environment is very complex, which requires very high computation and power consumption, as they require additional metric sensors. Also SLAM can create communication delays for real-time navigation. These problems have been taken care in further publications [22, 32]. Moreover, SLAM is predominantly a feature based method and its performance is not quite good for the indoor surfaces like walls/roofs, as the intensity gradient for these surfaces are very poor. A corridor is mostly made up of walls, roofs and floors, and hence the SLAM technique might not produce desirable navigation results.

Stereo Vision techniques like motion parallax, triangulation *etc.* are also used to estimate the depth and relative position using multiple cameras [3, 12]. Again, objects without prominent texture makes these techniques less reliable for real-time navigation. Other approaches use techniques like vanishing point [5, 23] estimation for autonomously navigating the UAVs in corridor environments.

In contrast to previous approaches, our approach will use an only a monocular camera to navigate autonomously without any 3D map generation. Hence, it is computationally very efficient. The proposed model doesn't depend on predefined path planning and hence, it tries to minimize the communication delay by fast processing the current frames. Our system is robust for avoiding walls during transit points. Our method will use the state-of-the-art deep learning techniques to navigate the UAV autonomously in indoor corridors. Deep learning is a trending area of research in the field of pattern recognition and machine learning. It refers to deep neural network based techniques that use supervised or unsupervised learning strategies, that automatically obtain hierarchical representations for various classification, recognition and regression tasks. The objective is to discover more abstract features in the higher levels of the representation, by using neural networks [30] which easily separates the various explanatory factors in the data. In the recent years, it has attracted much attention due to its state-of-the-art performance in diverse areas like object perception, speech recognition, computer vision, collaborative filtering and natural language processing.

ALVINN [24] showed how an artificial neural network (ANN) [30] mimics a human driver and efficiently performed autonomous navigation of UAVs. Ross *et al.* [28] proposed a machine learning strategy, the Dagger Algorithm [10], that can mimic a human pilot's actions in natural forest environments. In line with the above methods, we used a state-of-the art CNN model, DenseNet [14], to safely navigate the UAV in unknown corridor environments.

3. Proposed Method

The goal of our system is to enable a quadcopter to navigate autonomously in corridor environments, *i.e.*, to mimic the capabilities of a human pilot to take reasonable real-time decisions. The images are generated by a front-facing camera attached to the UAV. The trained classifier will return a real-time flight command for safe navigation. The classifier is trained with the expert pilot choice of actions. Our training process enables the classifier to learn most effective strategy to control the navigation with minimum probability of failure. We have used a state-of-the-art CNN model, DenseNet-161 [14], to train our model. Training of the model is done with supervised learning approach. The model parameters are learned by fine-tuning with our custom-made dataset. The input to the trained model is a real-time image generated from a camera attached to the front of the UAV. The output of the model will be the probability of different class labels, such as move forward, shift right, shift left and stop. If the confidence of the output is low then the model will again look into the next frame, else it will return a valid flight command to the UAV.

3.1. DataSet

Many datasets covering indoor scenarios [15, 25] are there. These publicly available datasets, however, are not useful in our method as not a single dataset contains ground truth values in terms of in-flight commands. Also, considering the fact that, manually labelling the available dataset with flight commands using inference increases the chance of error in the ground truth value, which can hamper the whole process of training, as the error in ground truth

certainly means erroneous experiment. Hence the creation of own dataset with the ground truth as flight command is needed to achieve the goal of the experiment.

Our custom dataset contains images that are captured with a front-facing static camera of the quadcopter from different lengths of the corridor. At each length, three different locations on a horizontal line cutting the corridor length perpendicularly, are selected. These locations are mainly the center and the two extreme sides of the corridor. At each location, three different images are captured by aligning the UAV camera as straight, left and right. Hence, we will have nine different images for a particular length of a corridor. Images are taken at different lengths of the corridor. A constant height (approximately $1m$) is sufficient for flying any quadcopter in any flat corridor. This assumption will keep the computations easy and also, it is feasible with respect to our experiment. We have collected a total of 3303 different images across 30 different corridors of varying dimensions. Total number of training images are increased to 13349 through various augmentation techniques, such as zooming, and flipping. During the creation of the dataset, it is made sure that all the locations of the corridor should have similar number of images. The dataset number break-up is shown in Table 1. The quadcopter can produce images of resolution 640×360, however for our model, resolution of 320×180 is sufficient enough to produce a reliable command.

In our real-time experiments, the quadcopter is controlled at a specific height using only four flight commands: Pitch-Forward, Roll-Left, Roll-Right and Stop. When the quadcopter is almost at the end of the corridor, the stop command is executed. For each indoor location, the quadcopter is controlled by using these four flight commands and observations are recorded with trial flights taken several times. We collected image frames that are captured from UAV front camera and manually labelled the flight commands corresponding to each frame. We have also covered as many as possible cases where the failure is prominent. For instance, consider a case when the quadcopter is too close to a wall. In this case, the quadcopter should land without any collision. Again, when it faces the walls to its left or right, it must try to move towards a direction opposite to the facing wall. The images that are captured from the UAV at runtime may contain noise. We manually discarded the noisy images from the dataset. Figure 2 shows the images of the corridor at nine different possible alignments of the UAV, on a single horizontal line perpendicular to the length of the corridor.

Table 1. Dataset Division

Sl.No.	Corridor Location	Number of Images	% of Images
1	Center of Corridor	4005	30
2	Left Side of Corridor	4672	35
3	Right Side of Corridor	4672	35

3.2. Training

We used the state-of-the-art, DenseNet-161 [14] pretrained model, which is the winner of the ImageNet Large Scale Visual Recognition (ILSVRC) [29] Challenge-2016. After removing the final layer of the original model, we augmented this model with 3 convolution layers followed by 1 fully connected (fc) layer at the end. The upper portion of the Figure 3 delineates the convolution layers of DenseNet-161 pretrained model, where as the lower portion layers are added by us. We trained the model with our custom dataset. We used Euclidean Loss (Equation 1) function to train our classifier. For fine-tuning, we decreased the learning rate by a factor of 2, if the loss remained same for 5 consecutive epochs. Initial learning rate is set to 0.001. We trained our model for 500 epochs with a mini batch of size 20. We used NVIDIA GTX 1080Ti GPU for training our model.

$$Euclidean\ Loss = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

where, y and $\hat{y}$ represent the ground truth and predicted class labels respectively, and N is the size of each mini batch used for training.

As shown in Figure 3, the model takes an input of size $180 \times 320 \times 3$ ($height \times width \times \#channels$) and produces four class labels. At any instant, the class with the highest probability is the desired flight command for navigation.

Fig. 2. Images of the corridor at nine different possible alignments of the UAV on a single horizontal line perpendicular to the length of the corridor. The first, second and third rows show the images, when the UAV is stationed at the left, right and center of the horizontal line, respectively. The first, second and third columns show the images, when the UAV is tilted to left, right and center, respectively.

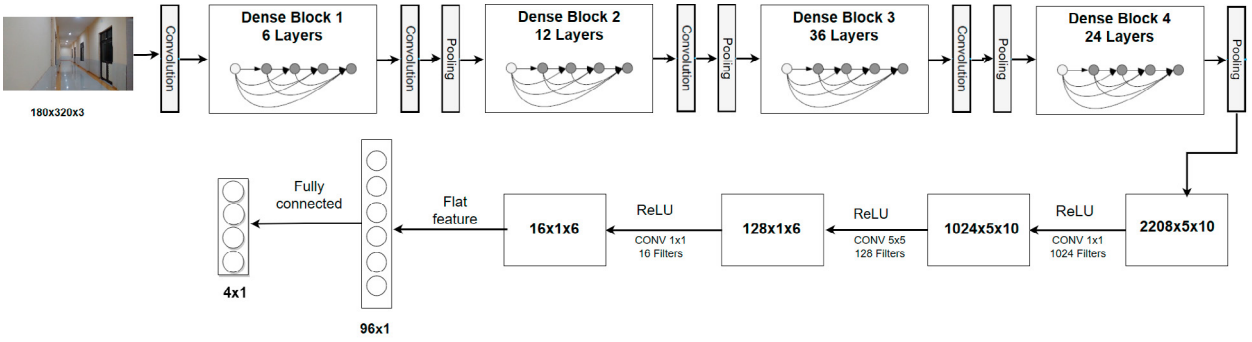

Fig. 3. Proposed Deep Neural Network which predicts the motion commands from images extracted from the UAV front camera

3.3. Navigation Algorithm

The UAV navigates autonomously in the corridor, until it receives a land command. Land command is generally issued at the end of the corridor. The complete set of rules for autonomous navigation in indoor corridor environments is delineated in Algorithm 1.

Algorithm 1: Autonomous UAV Navigation in corridors

Input: *img*: Image from the UAV front camera
Intialisation: *command* ← TAKE_OFF
while *command* ≠ *LAND* **do**
  position ← **Trained_Classifier**(*img*);
  if *position* = *Center* **then**
    Actuate UAV in PITCH_FORWARD direction;
  else if *position* = *Left* **then**
    Actuate UAV in ROLL_RIGHT direction to move towards the center;
  else if *position* = *Right* **then**
    Actuate UAV in ROLL_LEFT direction to move towards the center;
  else if *position* = *End* **then**
    send LAND command;
  img ← Extract the next frame from the UAV front camera;

4. Experimental Results

4.1. Hardware Platform

In our experiment, we have used a Parrot AR Drone (Figure 1) quadcopter as the UAV. It is equipped with a monocular front-facing static camera, a down-facing camera, an ultrasonic sensor for keeping track of ground altitude, a gyroscope and an accelerometer. The deep learning model is run on a host machine, which has Xeon(R) processor, 32GB RAM, NVIDIA GeForce GTX 1080Ti GPU running on Ubuntu 14.04 LTS. Communication between the UAV and the host machine is done via wireless LAN by using the Robot Operating System (ROS)[1] interface. Images are sent from the UAV to the host machine for further processing. Original images are captured at a resolution of 640×360. However, before passing the images through the trained neural network, the resolution is changed to 320×180. After processing the image though the trained CNN classifier, it generates a probable motion command. This command is then sent to the UAV for safe autonomous navigation.

4.2. Performance Evaluation

We check our model performance over all possible locations of the corridor. We followed the basic methodology for testing performance: a trial is said to be successful if the quadcopter after taking off, navigates safely to the end of the corridor and then lands safely. If the UAV is not able to fly the whole length of the corridor, it is considered to be a failure. Accordingly, we define two different accuracy measures to test our model performance.
(1) No-Collision-Ratio (NCR): Number of times over the total number of trials, the quadcopter successfully navigates the whole length of the corridor without any collision with the walls.
(2) Full-Flight-Ratio (FFR): Number of times over the total number of trials, the quadcopter successfully navigates the whole length of the corridor, albeit there may be slight side-wise collisions with the walls, which don't hamper the direction of the UAV.

Our model performance is delineated in Table 2. The overall accuracy across different corridors is calculated to be 0.773 and 0.847 in terms of NCR and FFR respectively. The accuracy can be improved in future with more number of training images from different corridors of varying dimensions.

[1] https://github.com/AutonomyLab/ardrone_autonomy

Table 2. Real-time Performance Evaluation in different corridor environments

Sl.No.	Corridors	#Trials	#Success (NCR)	#Success (FFR)
1	Corridor 1	50	35 (0.70)	39 (0.78)
2	Corridor 2	50	39 (0.78)	43 (0.86)
3	Corridor 3	50	42 (0.82)	45 (0.90)
TOTAL		150	116 (0.773)	127 (0.847)

5. Conclusion

Our proposed model uses a state-of-the-art CNN architecture, DenseNet-161, to classify the images extracted from a front-facing camera of a UAV. Based on classification result, the model generates a necessary control command to safely navigate the UAV in an unknown corridor environment. In contrast to some of the previous methods, our system only uses the inputs from a monocular camera. Hence it is computationally very efficient. We also present a dataset of images corresponding to different positions of the UAV through out different lengths of corridors, with varying dimensions. Also, the result of our navigation algorithm in real-world corridor environments is very encouraging. In future, the results can be enhanced by adding more number of training images from diverse corridor environments.

Acknowledgements

This research work has been partially supported by the following projects:
1. Grant Number 1(1)/ISEA-II/PMU/2015 of Information Security Education and Awareness (ISEA) Phase-II project funded by Ministry of Electronics and Information Technology (MeitY), Government of India.
2. Grant Number SB/FTP/ETA-0059/2014 by Science and Engineering Research Board (SERB), Department of Science and Technology, Government of India.

References

[1] Abbott, E., Powell, D., 1999. Land-vehicle navigation using gps. Proceedings of the IEEE 87, 145–162.
[2] Achtelik, M., Achtelik, M., Weiss, S., Siegwart, R., 2011. Onboard imu and monocular vision based control for mavs in unknown in-and outdoor environments, in: Robotics and automation (ICRA), 2011 IEEE international conference on, IEEE. pp. 3056–3063.
[3] Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N., 2009. Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments, in: Unmanned Systems Technology XI, International Society for Optics and Photonics. p. 733219.
[4] Bachrach, A., He, R., Roy, N., 2009. Autonomous flight in unknown indoor environments. International Journal of Micro Air Vehicles 1, 217–228.
[5] Bills, C., Chen, J., Saxena, A., 2011. Autonomous mav flight in indoor environments using single image perspective cues, in: Robotics and automation (ICRA), 2011 IEEE international conference on, IEEE. pp. 5776–5783.
[6] Blösch, M., Weiss, S., Scaramuzza, D., Siegwart, R., 2010. Vision based mav navigation in unknown and unstructured environments, in: Robotics and automation (ICRA), 2010 IEEE international conference on, IEEE. pp. 21–28.
[7] Bry, A., Bachrach, A., Roy, N., 2012. State estimation for aggressive flight in gps-denied environments using onboard sensing, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE. pp. 1–8.
[8] Çelik, K., Somani, A.K., 2013. Monocular vision slam for indoor aerial vehicles. Journal of electrical and computer engineering 2013, 4–1573.
[9] Checchin, P., Gérossier, F., Blanc, C., Chapuis, R., Trassoudaine, L., 2010. Radar scan matching slam using the fourier-mellin transform, in: Field and Service Robotics, Springer. pp. 151–161.
[10] Davies, W., Edwards, P., 2000. Dagger: A new approach to combining multiple models learned from disjoint subsets. machine Learning 2000, 1–16.
[11] Engel, J., Schöps, T., Cremers, D., 2014. Lsd-slam: Large-scale direct monocular slam, in: European Conference on Computer Vision, Springer. pp. 834–849.
[12] Fraundorfer, F., Heng, L., Honegger, D., Lee, G.H., Meier, L., Tanskanen, P., Pollefeys, M., 2012. Vision-based autonomous mapping and exploration using a quadrotor mav, in: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE. pp. 4557–4564.
[13] Huang, A.S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., Roy, N., 2017a. Visual odometry and mapping for autonomous flight using an rgb-d camera, in: Robotics Research. Springer, pp. 235–252.

[14] Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L., 2017b. Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, p. 3.
[15] Huitl, R., Schroth, G., Hilsenbeck, S., Schweiger, F., Steinbach, E., 2012. Tumindoor: An extensive image and point cloud dataset for visual indoor localization and mapping, in: Image Processing (ICIP), 2012 19th IEEE International Conference on, IEEE. pp. 1773–1776.
[16] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.
[17] McGuire, K., de Croon, G., De Wagter, C., Tuyls, K., Kappen, H., 2017. Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. IEEE Robotics and Automation Letters 2, 1070–1076.
[18] Mei, C., Sibley, G., Cummins, M., Newman, P., Reid, I., 2011. Rslam: A system for large-scale mapping in constant-time using stereo. International journal of computer vision 94, 198–214.
[19] Mellinger, D., Kumar, V., 2011. Minimum snap trajectory generation and control for quadrotors, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE. pp. 2520–2525.
[20] Mellinger, D., Michael, N., Kumar, V., 2012. Trajectory generation and control for precise aggressive maneuvers with quadrotors. The International Journal of Robotics Research 31, 664–674.
[21] Müller, M., Lupashin, S., D'Andrea, R., 2011. Quadrocopter ball juggling, in: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE. pp. 5113–5120.
[22] Nützi, G., Weiss, S., Scaramuzza, D., Siegwart, R., 2011. Fusion of imu and vision for absolute scale estimation in monocular slam. Journal of intelligent & robotic systems 61, 287–299.
[23] Padhy, R.P., Xia, F., Choudhury, S.K., Sa, P.K., Bakshi, S., 2018. Monocular vision aided autonomous uav navigation in indoor corridor environments. IEEE Transactions on Sustainable Computing .
[24] Pomerleau, D.A., 1989. Alvinn: An autonomous land vehicle in a neural network, in: Advances in neural information processing systems, pp. 305–313.
[25] Quattoni, A., Torralba, A., 2009. Recognizing indoor scenes, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE. pp. 413–420.
[26] Ramasamy, S., Sabatini, R., Gardi, A., Liu, J., 2016. LIDAR obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid. Aerospace Science and Technology 55, 344–358.
[27] Roberts, J.F., Stirling, T., Zufferey, J.C., Floreano, D., 2007. Quadrotor using minimal sensing for autonomous indoor flight, in: European Micro Air Vehicle Conference and Flight Competition (EMAV2007).
[28] Ross, S., Melik-Barkhudarov, N., Shankar, K.S., Wendel, A., Dey, D., Bagnell, J.A., Hebert, M., 2013. Learning monocular reactive uav control in cluttered natural environments, in: Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE. pp. 1765–1772.
[29] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115, 211–252.
[30] Schalkoff, R.J., 1997. Artificial neural networks. volume 1. McGraw-Hill New York.
[31] Sibley, G., Mei, C., Newman, P., Reid, I., 2010. A system for large-scale mapping in constant-time using stereo. International Journal of Robotics Research .
[32] Weiss, S., Achtelik, M.W., Chli, M., Siegwart, R., 2012. Versatile distributed pose estimation and sensor self-calibration for an autonomous mav, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE. pp. 31–38.