Pix2Pix-Based Depth Estimation from Monocular Images for Dynamic Path Planning of Multirotor on AirSim

Tomoyasu Shimada[1], Hiroki Nishikawa[1,2],
Xiangbo Kong[1], and Hiroyuki Tomiyama[1]

[1] Graduate School of Science and Engineering, Ritsumeikan University, Shiga, Japan
{tomoyasu.shimada, hiroki.nishikawa}@tomiyama-lab.org
{kong, ht}@fc.ritsumei.ac.jp
[2] JSPS Research Fellow

Abstract

Recently, autonomous flight for multirotor is actively researched. It is essential to get information about the real world for an autonomous flight from sensors. The sensors used by autonomous flight include a camera with a depth sensor or a stereo camera to generate a depth map. A distance measured by a depth camera depends on the performance of the camera. However, the high-performance camera is too heavy and expensive to load on a multirotor. Therefore, this paper proposes a method to generate depth maps by a monocular camera that is light and affordable. Besides, there are legally many limitations on an actual flight of a multirotor, and thus, autonomous flight in virtual environments is usually conducted in an early phase of development. To address this issue, this paper proposes a dynamic path planning method with collision avoidance using a monocular camera only and conducts simulation using AirSim. Experimental results show that the proposed method perfectly avoids collision.

1 Introduction

In recent years, multirotors are expected to play a variety of roles due to their convenience, and a large amount of research is being conducted on them. Examples of the roles include sports photography, aerial photography, infrastructure inspection, lifesaving, agriculture, and package delivery. Multirotors can also be used to monitor and search areas affected by fire, flood, and earthquake, which pose many risks that manned aircraft cannot. Unlike cars, multirotors can fly, making it possible to reach a destination in the shortest possible time, even if it takes a long time on the ground. Multirotors are also smaller in size than other aircraft such as helicopters and airplanes, allowing them to pass through narrow spaces. Therefore, it is considered to be suitable for carrying light loads, aerial photography in narrow alleys, and filming for sports broadcasts while the players keep moving. In order to take advantage of this convenience, research on autonomous multirotor flight is being actively conducted. For autonomous flight, it is essential to obtain information from sensors. A variety of sensors are used in the study of autonomous flight. For example, LiDAR and automatic dependant surveillance-broadcast (ADS-B) are used[1][2][3][4][5]. However, installing a large number of sensors, or high-performance sensors with a large weight, increases energy consumption. With the limited energy of a battery, long-distance flight becomes impossible due to the increased weight of sensors. Moreover, LiDAR and ADS-B are too expensive so the costs of experiments increases. Besides, in recent years, the laws and regulations for multirotors have become stricter, restricting the flight places and the flight speeds, etc., making it impossible to fully conduct experiments. In this paper, we use a flight simulator called AirSim in order to conduct experiments without being bound by laws and regulations. Cameras with a depth sensor (depth cameras) are more affordable than LiDAR and ADS-B. In this paper, autonomous flight is performed using only depth maps.

A lightweight depth camera that can be mounted on a multirotor can measure a distance of up to 10 meters. However, AirSim that is one of flight simulators using Unreal Engine 4 (UE4) can obtain the distance between the multirotor and the object from UE4. It can accurately obtain more than 200 meters and generate the depth map. There has been some research on deep learning of depth maps obtained by AirSim and images obtained from monocular cameras[6][7][8] [9][10]. This paper proposes a dynamic path planning method by using Pix2Pix. Pix2Pix generates a pair image, to generate an image close to the depth map that can be generated by AirSim from monocular images, and using the depth map. Subsection 3.1. shows the detail of Pix2Pix.

One of the vision-based methods is optical flow[11][12][13]. Optical flow-based methods can detect 2D vectors between frames. Therefore, it is effective in avoiding collisions in the lateral direction, but since it can only acquire two-dimensional vectors, the multirotor may not be able to avoid collisions with obstacles flying in front of it. On the other hand, there are some methods using depth maps[14][15][16]. the depth map-based methods detects the depth and thus avoid obstacles that face the multirotor in front. However, the depth cameras that can be mounted on real-world multirotors only accurately acquire distances up to 10 meters. The proposal method is collision avoidance of multirotor using the Pix2Pix model to generate a depth map approximately AirSim from a monocular image.

The rest of this paper is organized as follows. A method of path planning for collision avoidance is introduced in Section 2. Section 3 describes the overview of Pix2Pix and the details of algorithms of depth map generation from the monocular image. Section 4 shows the experimental results and Section 5 concludes this paper.

2 Path Planning for Collision Avoidance Using Depth Maps

In order to realize safe flight, multirotors are required to plan the path, which is to select the direction so that the multirotor can avoid colliding with objects. In this section, we describe a method for planning the path to avoid collision based on the state-of-the-art methods in [14][16]. The works in [14] and [16] introduce a method that divides a depth map into sections. In [14], the presented algorithm divides a depth map into five sections and selects the section which is the most distant object among them, as shown in Figure 1.

The presented method in [16], on the other hand, divides a depth map into 289 overlapped sections (17 rows and 17 columns) as shown in Figure 2 (a) and selects the most distant section as well as the method in [14]. The darkest place in the bounding box in Figure 2 (b) represents the farthest distance on the depth map.

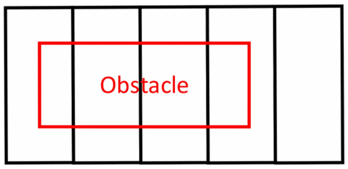
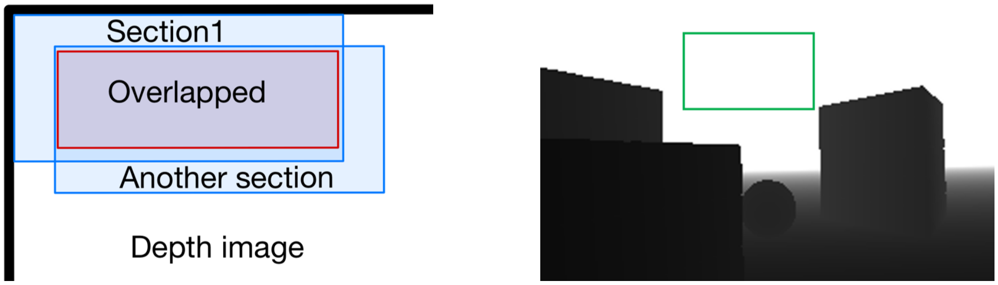


Figure 1: The method how to divide depth map Source[14]



(a) Overlapped section

(b) Section selection

Figure 2: The method how to divide and select section

In the real world, however, common depth cameras can hardly realize further distance than approximate 20 meters. The depth map may be useless in terms of collision avoidance since multirotors can fly approximate 20 meters per second at maximum and multirotors are difficult to suddenly fend off objects or to slow it down. Therefore, depth maps would be necessary to realize further distance of more than 20 meters for safe flight.

3 Pix2Pix-Based Generate Depth Map from Monocular Image

To overcome the issue presented in the prior section, this section introduces a method for a depth map from a monocular map based on Pix2Pix to realize distant places based on Pix2Pix.

3.1 Introduction of Pix2Pix

Pix2Pix is an image generator based on cGAN (Conditional Generative Adversarial Networks) in [17]. Figure 3 shows the overview of cGan in Pix2Pix. Generator in cGAN generates an image from the input image and conditional noise. The GAN consists of two networks, Generator and Discriminator, as shown in Figure 3. Generator learns to prevent the generated image from being detected as the one generated by the discriminator, and the discriminator learns not to misidentify the generated data from the training data. The Discriminator learns not to misidentify the training data and the generated data and finally generates an image similar to the training data. In addition, Pix2Pix uses U-NET[18] as Generator, PatchGAN as Discriminator.

3.2 Depth Map Generation from Monocular Image

In the real world, we can generate accurate depth maps of only 10 to 20 meters at most. This means that collision avoidance using depth maps is inaccurate and slow flight is unavoidable. However, AirSim, the simulator used in this research, can obtain accurate depth maps up to 200 meters or more due to it obtains information from UE4. Therefore, a proposed method is generating depth maps from monocular images by learning on Pix2Pix as a pair of monocular images and depth maps acquired by AirSim. Figure 4 shows the monocular image, a depth

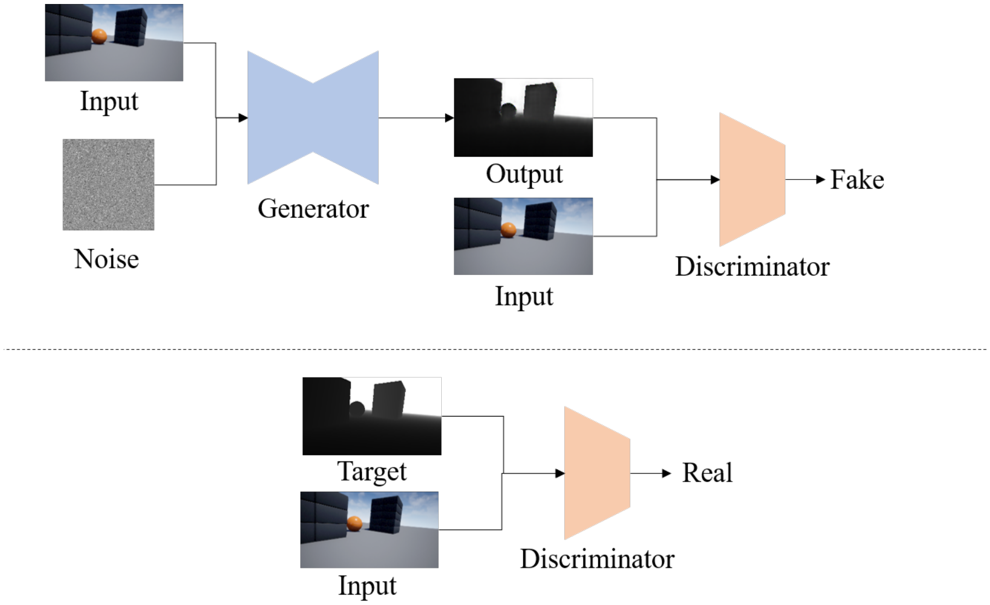
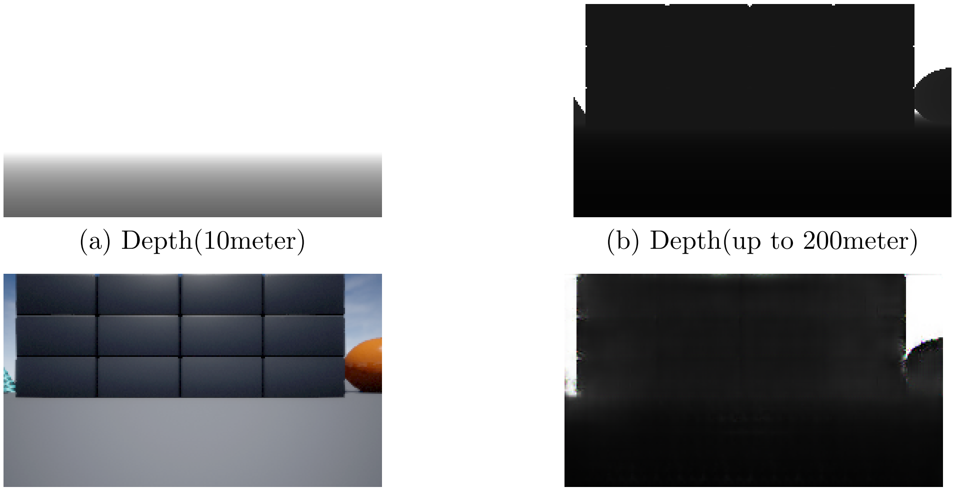


Figure 3: cGAN in Pix2Pix

map to 10 meters and a depth map up to 200 meters taken at the same location, and an image generated by Pix2Pix.


(a) Depth(10meter)

(b) Depth(up to 200meter)

(c) monocular

(d) Pix2Pix

Figure 4: Comparison of Depth Maps

To output an image like (d) with the input of (c) in Figure 4, a Generator trained by cGAN is used. The Generator uses a network called U-net, which has an encoder-decoder structure. In the encoder, features are extracted by convolutional and pooling layers, and in the decoder, features are preserved by convolutional and up-sampling layers, and the image size can be restored to a larger size. By inputting the input image, a monocular image, into Generator, an image like (d) in Figure 4 can be output. In this way, a depth map is generated from the monocular image and combined with the collision avoidance algorithm described in Section 2 to achieve highly accurate collision avoidance. The next section describes an experiment to compare the accuracy of the images generated by Pix2Pix and the collision avoidance algorithm.

4 Experiment

This section shows Pix2Pix training results and experiments conducted to verify the performance of the images produced by Pix2Pix.

4.1 Learning on Pix2Pix for Generation of Depth Map and Comparison of Similarity of Images

This section describes a learning environment of Pix2Pix and a loss function during learning, and a result of a comparison of the similarity of images. The specifications of the computer on which we were training are shown in Table 1. The learning conditions are also shown in Table 2.

Table 1: Environment of learning

OS	Windows 10 pro
RAM	32GB 2666MHz
CPU	Intel Core i7-9700K 3.60GHz
GPU	NVIDIA GeForce RTX 2070super 8GB
Location	Blocks (AirSim Binary version)

Table 2: Condition of learning

Epochs	500
Pairs	10000
Batch Size	1

The computer used for the study is Windows 10Pro OS, 32GB RAM, Intel Corei7-9700K CPU, NVIDIA GeForce RTX 2070Super GPU, and the map used is Blocks, a binary version of AirSim. Figure 5 shows an overhead view of Blocks. The training conditions were as follows: the number of Epochs is 500 since the loss function became smoother when Epochs exceeded 400. The number of pairs between monocular images and depth maps obtained by AirSim is 10,000. Batch size is 1 due to in general, using small batch size gives higher SSIM(structural similarity).

The experiment describes the comparison of the similarity of the images between Pix2Pix and depth maps. The images of the location where the image is taken, the depth map obtained, and the image generated by Pix2Pix are shown in Figure 5.

In order to quantify the similarity of these images, PSNR (Peak signal-to-noise ratio) and SSIM are used. PSNR is obtained by the following equation.

$$PSNR = 10\log_{10}\Big(\frac{255^2}{MSE}\Big) \tag{1}$$

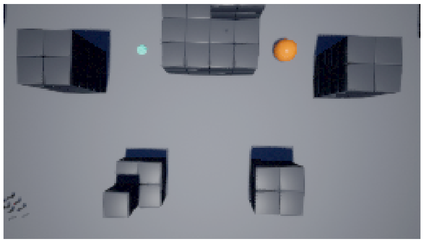


Figure 5: Overhead View of Blocks

MSE (Mean Squared Error) is obtained by the following equation.

$$MSE = \frac{1}{ColorN} \sum_{i=0}^{Color} \sum_{j=1}^{N} E(i,j)^2 \tag{2}$$

PSNR indicates how much the pixel brightness value at the same location has changed. PSNR is also considered to be acceptable at 30 dB or more. Therefore, in this paper, we will use 30 or more as a standard. SSIM is a measure that takes the average, variance, and covariance of surrounding pixels based on brightness, contrast, and structure, and incorporates correlations with surrounding pixels as well as individual pixels. SSIM is obtained by the following equation.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

Table 3 shows a table in which 10000 images used as a data-set are evaluated using these indices. Table 4 shows a table in which 10000 images not used as a data-set.

Table 3: Accuracy of Images used as a data-set

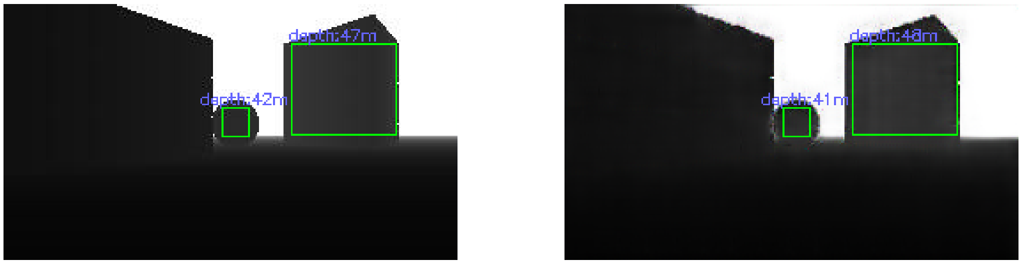
	PSNR(dB)	SSIM
maximum	34.066	0.986
minimum	17.816	0.871
average	29.436	0.970

Table 4: Accuracy of Images not used as a data-set

	PSNR(dB)	SSIM
maximum	34.099	0.985
minimum	17.878	0.883
average	28.648	0.966

Table 3 and 4 show that the value of the average PSNR of Blocks depth image is under 30 dB and the value of average SSIM is approximate 0.96. The value of PSNR is below the target of 30 dB, but the value of SSIM is high at 0.9. Figure 6 shows the difference of depth value between a depth map and a output image from Pix2Pix model.

As shown in Figure 6, the depth of the sphere in the center is 42 meters, and the output of Pix2Pix is 41 meters. The depth of the building on the right is 47 meters, and the output of



(a) Depth map

(b) Image generated by Pix2Pix

Figure 6: The Difference of Depth Value

Pix2Pix is 48 meters. Therefore, the error is about 1 m, which is not a problem for collision avoidance. This will be confirmed in the next experiment.

4.2 Comparison of Collision Avoidance by Each Method with Pix2Pix

This section presents the results of a comparison between two collision avoidance methods and three image acquisition methods, for a total of six different methods. Collision avoidance methods Ma Method[14] and Prez Method[16] are used, and three methods of image acquisition are used: a realistic 10-meter depth map, a 200-meter depth map obtained by AirSim, and images obtained by Pix2Pix. Figure 7 shows the processing of each method.

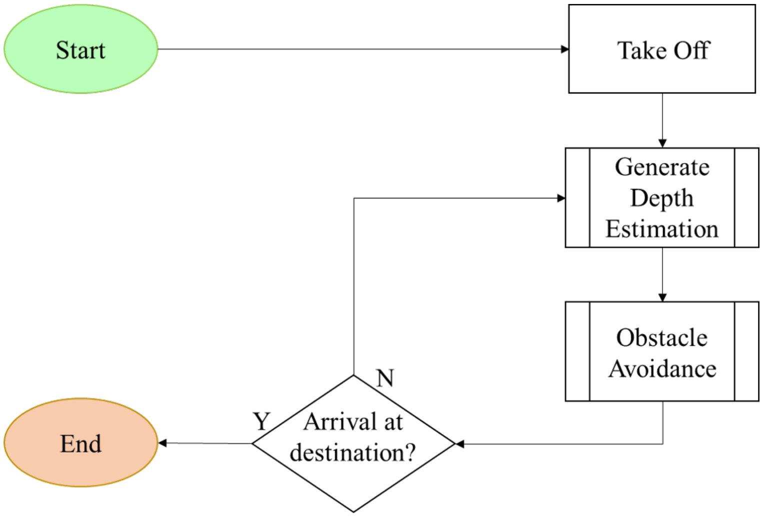


Figure 7: Comparison of depth maps

Firstly, this system gets a depth map. Secondly, this system divides the depth map and selects the best section. Finally, This system will control the velocity of the multirotor so that the selected section is in the center. Each method is implemented by modifying the section selection method and the depth map acquisition method.

The experiment will be conducted with Blocks and the speed in the direction of travel will be set to 3 m/s. The results of 100 flights to random coordinates 100 meters from the starting point will be summarized. The evaluation criteria are the collision rate and processing time. The processing time of the experiment is the time from image acquisition to velocity control in this flowchart. The experimental results are shown in Table 5.

Table 5: Comparison of Each Method

	collision rate (%)	processing time (s)
10m-Ma	70.707	0.071
10m-Prez	70.707	0.356
200m-Ma	63.636	0.072
200m-Prez	2.062	0.359
Pix2Pix-Ma	76.699	0.045
Pix2Pix-Prez	0.000	0.525

The experimental results show that Ma Method has an overall faster processing time than the Prez method and a higher collision rate than the Prez method. This method does not allow collision avoidance upward, while the multirotor allows collision avoidance upward. As a result, there are many cases in which the multirotor fails to avoid obstacles. On the other hand, the Prez Method has a longer processing time than the Ma Method, but a lower collision rate than Ma method. In particular, the collision rate of Pix2Pix-Prez is lower than 200m-Prez, since it takes time for the image to be updated when using Pix2Pix, therefore the collision can be avoided by flying high enough to reach a height where there are no obstacles.

5 Conclusion

This paper presents the use of Pix2Pix to obtain highly accurate depth maps to avoid multirotor collisions. The collision rate of the proposed method is 0, over-performing the related works. Even when Pix2Pix is used, the results showed that there were few collisions. In order to implement the system on a real multirotor, it is necessary to install a high-performance computer. Our future work is to study and experiment on how to increase the speed of the system so that it can be used in actual multirotors. The investigation of generalization performance is also a future task.

Acknowledgment

This work is partly supported by KAKENHI 20K23333 and 20J21208.

References

[1] Florent Martel, Richard Schultz, Ziming Wang, Mariusz Czarnomski, and William Semke, "Unmanned Aircraft Systems Sense and Avoid Avionics Utilizing ADS-B Transceiver," in *Aerospace Conference and AIAA Unmanned... Unlimited Conference*, 2009.

[2] Subodh Bhandari, Nicole Curtis-Brown, Isaac Guzman, Tristan Sherman, Joshua Tellez, and Edward Gomez, "UAV Collision Detection and Avoidance using ADS-B Sensor and Custom ADS-B Like Solution," in *AIAA Information Systems-AIAA Infotech@ Aerospace*, 2017.

[3] Joshua Redding, Jayesh Amin, Jovan Boskovic, Yeonsik Kang, Karl Hedrick, Jason Howlett, and Scott Poll, "A Real-Time Obstacle Detection and Reactive Path Planning System for Autonomous

Small-Scale Helicopters," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007.

[4] Andrew Moffatt, Eric Platt, Brandon Mondragon, Aaron Kwok, Dennis Uryeu, and Subodh Bhandari, "Obstacle Detection and Avoidance System for Small UAVs Using A LiDAR," in *International Conference on Unmanned Aircraft Systems*, 2020.

[5] Yawei Hou, Zhenling Zhang, Chao Wang, Shouhu Cheng, and Demao Ye, "Research on Vehicle Identification Method and Vehicle Speed Measurement Method Based on Multi-rotor UAV Equipped with LiDAR," in *International Conference on Advanced Electronic Materials, Computers and Software Engineering*, 2020.

[6] Shaoyong Zhang, Na Li, Chenchen Qiu, Zhibin Yu, Haiyong Zheng, and Bing Zheng, "Depth Map Prediction from A Single Image with Generative Adversarial Nets," *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 14357–14374, 2020.

[7] L. Madhuanand, F. Nex, and M. Yang, "Deep Learning for Monocular Depth Estimation from UAV Images," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 451–458, 2020.

[8] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2024–2039, 2016.

[9] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "J-MOD2: Joint Monocular Obstacle Detection and Depth Estimation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1490–1497, 2018.

[10] Kyle Hatch, John Mern, and Mykel Kochenderfer, "Obstacle Avoidance Using a Monocular Camera," in *in AIAA Scitech Forum*, 2021.

[11] Zhi Hou, Juntong Qi, and Mingming Wang, "Fusing Optical Flow and Inertial Data for UAV Motion Estimation in GPS-denied Environment," in *Chinese Control Conference*, 2019.

[12] Dong-Wan Yoo, Dae-Yeon Won, and Min-Jea Tahk, "Optical Flow Based Collision Avoidance of Multi-rotor UAVs in Urban Environments," *International Journal of Aeronautical and Space Sciences*, vol. 12, no. 3, pp. 252–259, 2011.

[13] Haiyang Chao, Yu Gu, and Marcello Napolitano, "A Survey of Optical Flow Techniques for UAV Navigation Applications," in *International Conference on Unmanned Aircraft Systems*, 2013.

[14] Chenxiang Ma, You Zhou, and Zhiqiang Li, "A New Simulation Environment Based on AirSim, ROS, and PX4 for Quadcopter Aircrafts," in *International Conference on Control, Automation and Robotics*, 2020.

[15] Johann Borenstein and Yoram Koren, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," *IEEE transactions on robotics and automation*, vol. 7, no. 3, pp. 278–288, 1991.

[16] Erwin Perez, Alexander Winger, Alexander Tran, Carlos Garcia-Paredes, Niran Run, Nick Keti, Subodh Bhandari, and Amar Raheja, "Autonomous Collision Avoidance System for a Multicopter using Stereoscopic Vision," in *International Conference on Unmanned Aircraft Systems*, 2018.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *in IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015.