robotics

MDPI

Article

Reduced Simulation: Real-to-Sim Approach toward Collision Detection in Narrowly Confined Environments

Yusuke Takayama [1,†], Photchara Ratsamee [1,2,†] and Tomohiro Mashita [1,2,*,‡]

[1] Graduate School of Information Science and Technology, Osaka University, 1-5, Yamadaoka, Suita 565-0871, Japan; takayama.yusuke@lab.ime.cmc.osaka-u.ac.jp (Y.T.); photchara@ime.cmc.osaka-u.ac.jp (P.R.)
[2] Cybermedia Center, Osaka University, 5-1, Mihogaoka, Ibaraki 567-0047, Japan
* Correspondence: mashita@ime.cmc.osaka-u.ac.jp
† These authors contributed equally to this work.
‡ Current address: 1-32 Machikaneyama, Toyonaka, Osaka 560-0043, Japan.

Abstract: Recently, several deep-learning based navigation methods have been achieved because of a high quality dataset collected from high-quality simulated environments. However, the cost of creating high-quality simulated environments is high. In this paper, we present a concept of the reduced simulation, which can serve as a simplified version of a simulated environment yet be efficient enough for training deep-learning based UAV collision avoidance approaches. Our approach deals with the reality gap between a reduced simulation dataset and real world dataset and can provide a clear guideline for reduced simulation design. Our experimental result confirmed that the reduction in visual features provided by textures and lighting does not affect operating performance with the user study. Moreover, by conducting collision detection experiments, we verified that our reduced simulation outperforms the conventional cost-effective simulations in adaptation capability with respect to realistic simulation and real-world scenario.

Keywords: reduced simulation; collision detection; micro aerial vehicles

check for updates

Citation: Takayama, Y.; Ratsamee, P.; Mashita, T. Reduced Simulation: Real-to-Sim Approach toward Collision Detection in Narrow Confined Environment. *Robotics* **2021**, *10*, 131. https://doi.org/10.3390/robotics10040131

Academic Editors: Stefan Gheorghe Pentiuc and Marco Ceccarelli

Received: 3 August 2021
Accepted: 2 December 2021
Published: 8 December 2021

1. Introduction

Micro aerial vehicles (MAVs) have recently cultivated a large market of industrial applications, e.g., infrastructure inspection, monitoring power transmission lines or surveillance, delivery, and emergency response [1–3]. Apart from broad liberated situations, MAVs are also used in narrow or confined environments (e.g., vibration isolation damping, underground pits, ceilings, and piping networks). However, an MAV still needs to be operated by an experienced pilot using a remote control. Due to the shortage of pilots, the service cannot be scaled to satisfy the potentially huge market demand. Therefore, autonomous flight technology in narrow or confined spaces is required.

Recent advancements in autonomous flight technologies have realized non-collision motion planning in corridors and on city roads without a GPS signal or advanced sensing (e.g., light detection and ranging) [4–6]. There have also been recent and rapid advances taking place in convolutional neural networks (ConvNets) and reinforcement learning, which have adopted deep learning models, where the quality of datasets determines performance. Some of these studies [4,5] use data collected from the real environment. However, the situation in which we can obtain affluent real-world data is rare.

This fact motivates researchers to use a dataset from simulations to tackle simulation-based learning methods. Due to the differences between the real world and simulation, termed *reality gap*, the problem remains unsolved. The simulation-to-reality (sim-to-real) approach, which aims to improve the fidelity of simulations, is a promising one. However, visual sim-to-real still requires a large, diverse, and high-fidelity dataset [7,8]. These datasets consist of scanned real-world data [7] or synthetic models [8]. In a narrow or confined space, retrieving a 3D model from scanned real-world data is a difficult task due to limitations placed on the movement of workers from entering the area. Additionally, a synthetic dataset requires a task-specific bespoke model, which may result in a significant cost increase. Therefore, the conventional sim-to-real approach is not suitable for industrial applications in narrow or confined environments. Moreover, due to the limitations of this approach, in a recent symposium, researchers have discussed the value of using sim-to-real [9].

With respect to the background literature on the reality to simulation (real-to-sim) approach, a complementary methodology to the sim-to-real has appeared [10–12]. This approach converts real-world image data into a simulation. However, conventional methodologies employ the real-to-sim approach as only part of the bilateral style translation. Therefore, manual engineering and adjustment of deep models cause problems in task-specific industrial applications. Moreover, the current studies on real-to-sim target the grasping task of the manipulator [13]. These factors motivated us to embed non-learning traditional image processing into a real-to-sim approach for MAV applications.

One important insight derived from this question: ***What is the essential or minimal visual information required to realize MAV visual control?*** In this paper, we propose a concept of *reduced simulation*, which is a data generation scheme for training a machine learning model. *Reduced simulation* does not require too much information from two sources, saving computational costs in a simulation and eliminating unnecessary information from a real image, respectively. We tested some rendering styles, which gradually decreased the number of visual features through user studies on MAV flights in simulated narrow or confined environments. We evaluated our *reduced simulation*, which has limited visual features, on MAV collision detection in real environment. Our contributions are as follows:

- We tackled a collision detection problem in narrow or confined environments with a novel real-to-sim concept. This restricted environment has not been addressed in previous studies.
- We confirmed that the rendering style of reduced simulation does not reduce the performance of MAV control through the subject experiment.
- We evaluated our pipeline by carrying out similar experiments as those carried out in previous studies [4] on real experiment sites of the ceiling environments. By conducting experiments, we confirmed that our reduced simulation pipeline outperformed, within the adaptation capabilities, the traditional cost-saving simulation technique.
- Based on the results of our experiment, we provided guidelines for adopting a cost-saving reduced simulation for MAV collision detection in cluttered environments.

2. Related Works

2.1. Vision Based MAV Control

Before the deep learning era, previous studies utilized traditional visual features, such as optical flow or vanishing point [14,15]. These studies revealed the possibility of incorporating low-level visual features for MAV control. However, the dependency on manual engineering has hindered the subsequent development of the aforementioned approach. Some prior studies have also utilized reinforcement learning [16,17]. Similar to our study, the literature [16] used a human operator as supervised data. However, narrow or confined environments are not suitable for imitation learning because complex situation recognition is required. ConvNet has been greatly utilized as another type of learning-based method [4–6,18]. Regardless of whether a data source is a simulation or a real-world environment, these sources are difficult to handle in narrow or confined contexts because of the dependency on large-scale and realistic datasets.

2.2. Sim-to-Real Approaches

The learning-based methodology for visual control suffers from the limitation of the reality gap. The recent mainstream method for handling the reality gap is the sim-to-real approach. Most of the conventional datasets employ 3D scanned data of real-world environments [7,19,20].

The simulators based on these datasets are also reported [10,21]. These datasets and simulators have supported visual control research in indoor environments, such as corridors and ordinary houses. However, these datasets do not contain a model of a narrow or confined environment, and creating a task-specific dataset of narrow or confined environments from real-world scanning is impractical. Additionally, simulators based on synthetic data have already been developed [8,22,23]. These datasets can easily be expanded by adding 3D models. However, off-the-shelf synthetic models of narrow or confined environments are not cost effective. Thus, it is not appropriate to use the existing sim-to-real type datasets and simulators or conventional methods directly for industrial applications in narrow or confined environments. Previous research [24] has handled this gap between real-world-based simulators. This study has revealed the effectiveness of splitting visual perception and motion control. However, synthetic simulators and the real world were not targeted.

2.3. Real-to-Sim Approach

A new research paradigm handling the reality gap is the real-to-sim approach. Zhang et al. [11] investigated a real-to-sim domain adaptation for visual control. Their study also emphasized the advantages of decoupling visual perception and motion control. However, this study did not provide guidelines for designing simulated environments. Moreover, conventional works have not applied the real-to-sim concept to MAV's visual control. Therefore, we attempted to apply real-to-sim to a MAV flight in narrow or confined environments by including design principles for simulation.

3. Reduced Simulation Concept

The critical insight of our approach is simple. The cost problem and visual reality gap results depending on the diversity of texture, lighting effect, or minor objects in the real world. Do these features also affect the maneuver itself? If not, we should omit these visual features in order to simplify the simulation. The real-to-sim approach essentially aims to transfer real-world data into simulation data. However, conventional studies have utilized real-to-sim transfer only as a complementary method to sim-to-real transfer. Thus, we tried to return to the original concept of real-to-sim. We illustrated this concept in Figure 1.

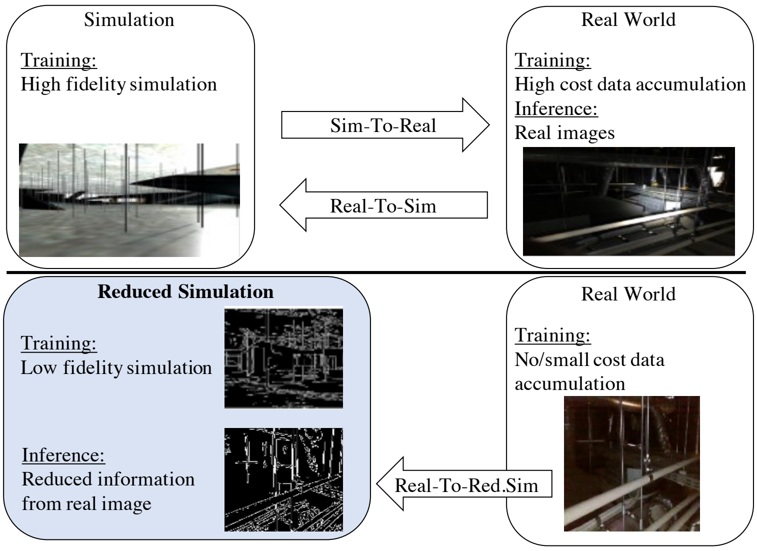


Figure 1. The concept of reduced simulation (top), conventional real-to-sim approach (bottom), and the proposed real-to-reduced simulation (real-to-red.sim) approach. Our proposed concept is that a machine learning model is trained with a dataset generated by a *reduced simulation* and then inferred with *real-to-red.sim* data from real images using image processing. By comparing the conventional real-to-sim, which uses CycleGAN [25] as the main component, our proposed approach includes only simple image processing, such as canny edge extraction.

One of the issues concerning real-to-sim is the lack of guidance in designing the simulation. Previous research [18] has examined the adaptation capability of simple simulations. However, this study did not provide any design guidelines. In order to establish guidelines for the reduced simulation, we focused on human visual cognition: The ability of human vision to recognize a 3D structure from a single or edge image has been verified in the fields of art, psychology, and cognitive science [26]. Importantly, the linear perspective technique does not comprehend some visual characteristics, such as texture and lighting. We doubted the necessity of incorporating abundant visual features, which a conventional visual perception approach includes, for 3D structure cognition. Moreover, this is the same in a task for automatic MAV control based on visual perception. Therefore, we proposed a novel concept, *reduced simulation*, which combined this insight and the conventional learning-based approach.

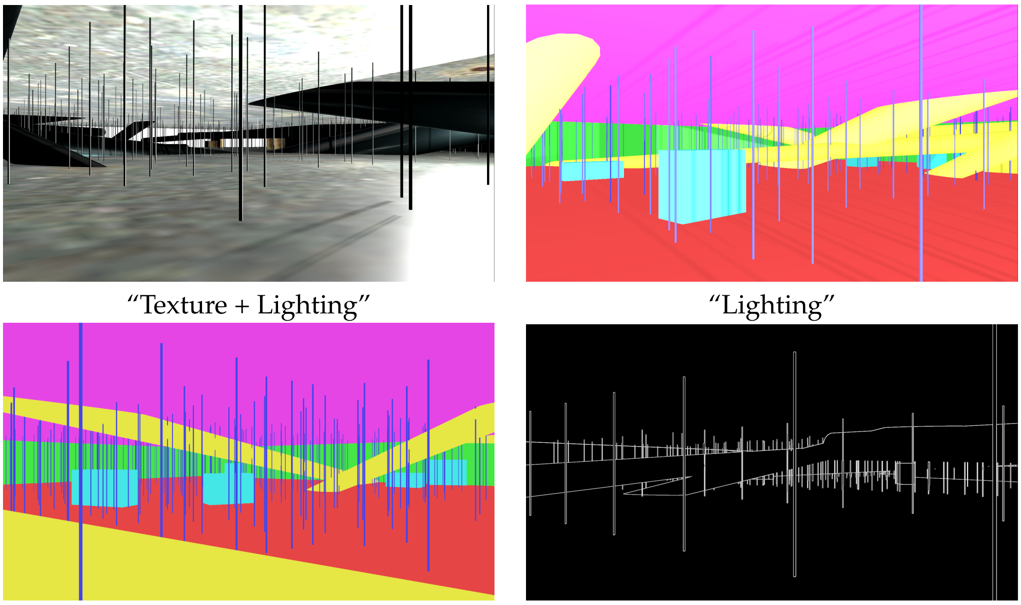
Our proposed concept can also function as the methodology to make the learning of motion control independent from visual perception [24]. In other words, we required the real-to-red.sim transfer to be applicable with the trained deep model by reduced simulation into practical usage. In this paper, we considered both non-learning traditional image processing and learning-based instance segmentation [27,28] as this transfer. When compared to edge extraction by using traditional image processing, instance segmentation provides additional information: What type of objects does the edge belong to? If this extra information is necessary, we can adapt the instance segmentation method for industrial applications in narrow or confined environments. Therefore, we tested its necessity by using user study and experiments.

4. User Study for Reduced Simulation

In the user study, we evaluated the performance of test subjects operating a MAV in four types of simulation environments. As the application scenario for operating the MAV, the MAV had to perform surveillance of a space behind a ceiling, which was a surveillance task carried out before any maintenance or replacement work of air conditioners in the building. We asked participants to fly MAV in order to maximize the coverage area of the environment that the participant surveyed within the limited time. The assumptions and purposes of operating the MAV were explained to the subjects.

We prepared simulation environments comprising metal boxes, which contained electrical equipment, pipes (including air ducts), and hanging bolts. In order to avoid applying the same environment twice to one subject, the objects were randomized according to the variations indicated in Table 1. We placed four light sources on the floor that simulated lighting from maintenance holes. The positions of the light sources were also randomized. All objects in the environment had collision detection. In one environment, when the collision happens, the robot will respawn at the start point without changing the environment. The objective is to record the total coverage area the operator can cover within the limited time. When the MAV collided, it respawned at the origin of the environment and exploration continued. The size of the MAV was 120 mm × 100 mm × 20 mm, and the camera on the MAV had an 82° viewing angle.

We implemented four types of simulation environment: texture + lighting, lighting, color segmentation, and edge extraction, as shown in Figure 2. In order for subject to become familiar with MAV control, they are asked to take nine practice flights before the actual experiment. During the actual experiment, a subject operated the MAV three times for each type of simulation environment. In order to avoid bias in growing experience, participants operate MAV in random orders decided by Latin square design [29] in order to counterbalance immediate sequential effects.



“Texture + Lighting”

“Lighting”

“Color Segmentation”

“Edge Extraction”

Figure 2. Examples of simulations.

Table 1. Size variation of the objects in the environment.

Type	Number	Scale (x, y, z)
Box	20	(0.2~0.8, 0.2~0.6, 0.2~0.8)
Hanging Bolt	800	(0.01, 1, 0.001)
Pipe	10	(0.2~0.8, 1000, 0.2~0.8)

4.1. Hypothesis

We will confirm the following hypothesis: The reduction in visual effect by texture and lighting has no effect on the operating performance evaluated with the number of collisions and exploration area. If this hypothesis is supported, collision detection can be achieved from reduced images.

4.2. Experiment Setup

The simulation setup used for our experiment comprised a MAV simulator based on the ROS and Unity renderer. The input device for operating MAV was Sony DualShock4.

We implemented the MAV simulation setup by referring to Meyer et al. [30]. In order to apply the parameters of the MAV simulator to the small MAV that we assumed, we modified the value of the principal moment of inertia of the 120 mm× 100 mm × 20 mm box that weighed 200 g. The max speed was 3 m/s. A proportional controller was used. Unity received MAV behavior from the ROS, rendered the graphics of the environment, and detected a collision.

MAV behavior during the experiment was logged every 0.1 s. The area of observation was calculated following the experiment. The number of subjects was 24 (8 females and 16 males between 21 and 45 years of age). The average age of the subjects was 25. Four subjects had some experience operating a MAV that weighed less than 200 g, and one subject had experience operating a MAV that weighed over 200 g.

4.3. Comparison among Types of Simulation

Figure 3 shows the histogram of coverage by each simulation style. Table 2 shows the result of the Shapiro–Wilk test. The findings suggested that lighting did not fit the Gaussian distribution. We then conducted the Friedman test, and the result was $p = 0.4843$.

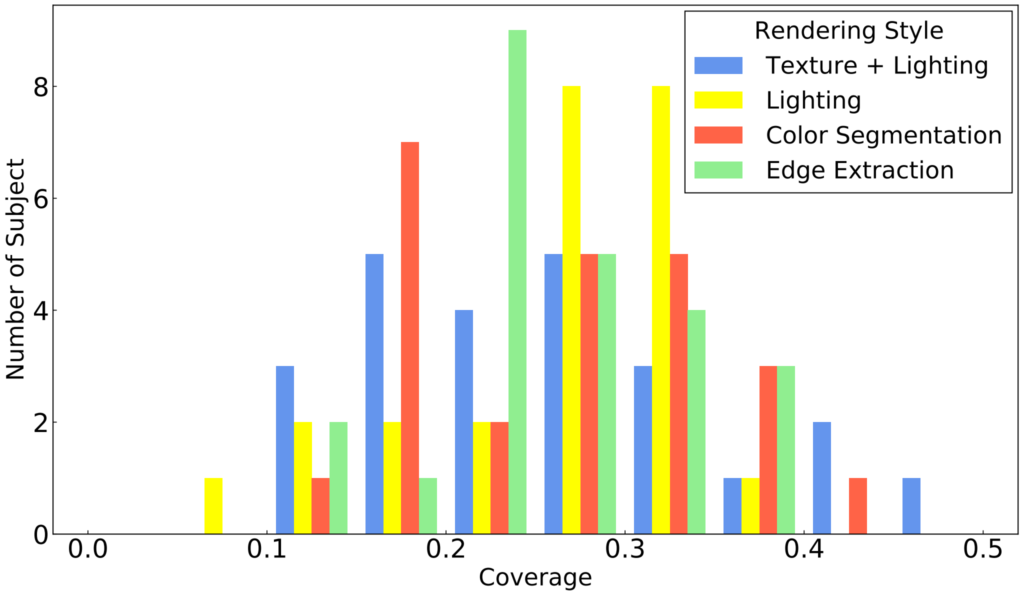


Figure 3. An histogram of the coverage distribution for each drawing style. Apart from the lighting-only style, these distributions follow a normal distribution. The results proved that there was no statistical significance between the groups beyond individual differences.

Table 2. Result of Shapiro–Wilk normality test.

	Texture + Lighting	Lighting	Color Segmentation	Edge Extraction
p-value	0.3776	0.0353	0.2228	0.62977

After examining both these results, no significant difference was observed between the groups in terms of simulation style. We also conducted the Friedman test on the data obtained from the number of collisions. The result of the test was $p = 0.4630$. No significant difference was again observed in terms of the simulation style between the groups. Although these results are of no significant difference, the tests cannot completely verify the equal performance in simulation styles, and this result shows a high possibility for our hypothesis. We, therefore, believe that our reduced simulation is a reasonable concept.

5. Collision Detection with Reduced Simulations

In this section, we present the application of our concept of reduced simulation to train a Neural Network (NN) model and to test with the real environment. Figure 4 shows the overall process of training and testing. Firstly, we generate a low-cost dataset from reduced (low cost) simulation. Afterward, we train the NN model with images obtained from a low-cost dataset. Finally, we apply the NN model to reduced images converted by an image processing technique from real environment images, and the NN model detects a collision.

5.1. Dataset Generation

In order to train and test the classification model, we collected data on images when the MAV traveled into free space and also at the time of collision. We prepared three datasets: low-fidelity simulation, moderate-fidelity simulation, and the real world. Sample images taken from the simulation datasets are depicted in Figure 5. The size of images obtained from the simulation is 576 × 256 pixels. In order to adjust the input size of the NN model, all of the images were cropped to 256 × 256 pixels from the image center. In order to create simulation datasets, we adapted the bite-the-bullet method used in the pioneering study by [4]. Our virtual MAV was spawned in randomly sampled positions in the Unity environment and was then moved straight until it collided with an object. Figure 6 illustrates the path taken by MAV during data collection. Through this phase, for each type of object, we used three texture images as shown in Figure 7.

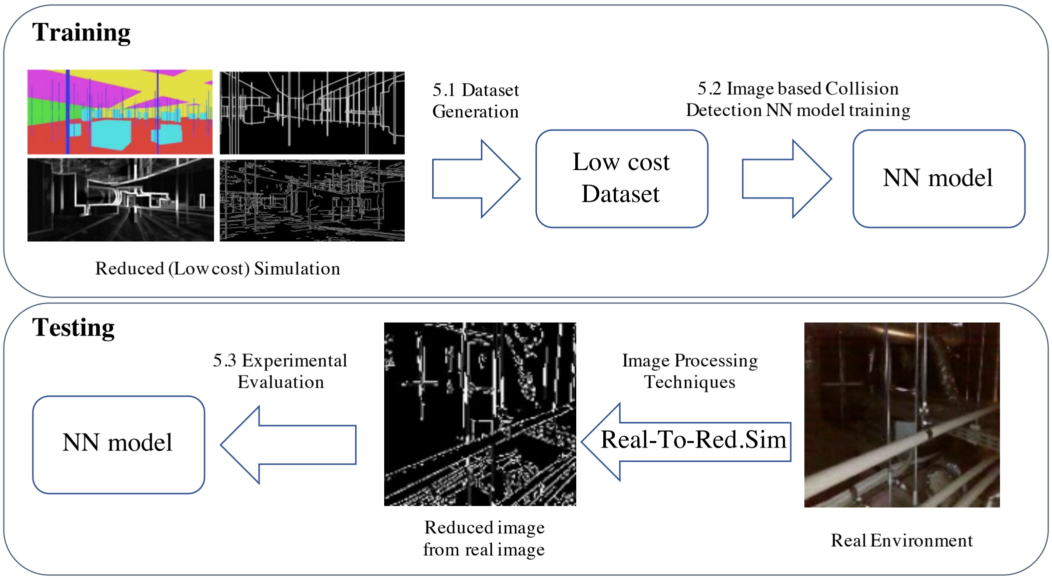

Training
Reduced (Low cost) Simulation
5.1 Dataset Generation
Low cost Dataset
5.2 Image based Collision Detection NN model training
NN model
Testing
5.3 Experimental Evaluation
NN model
Reduced image from real image
Image Processing Techniques
Real-To-Red.Sim
Real Environment

Figure 4. The overall process of training and testing for the experiment of collision detection with reduced simulation.

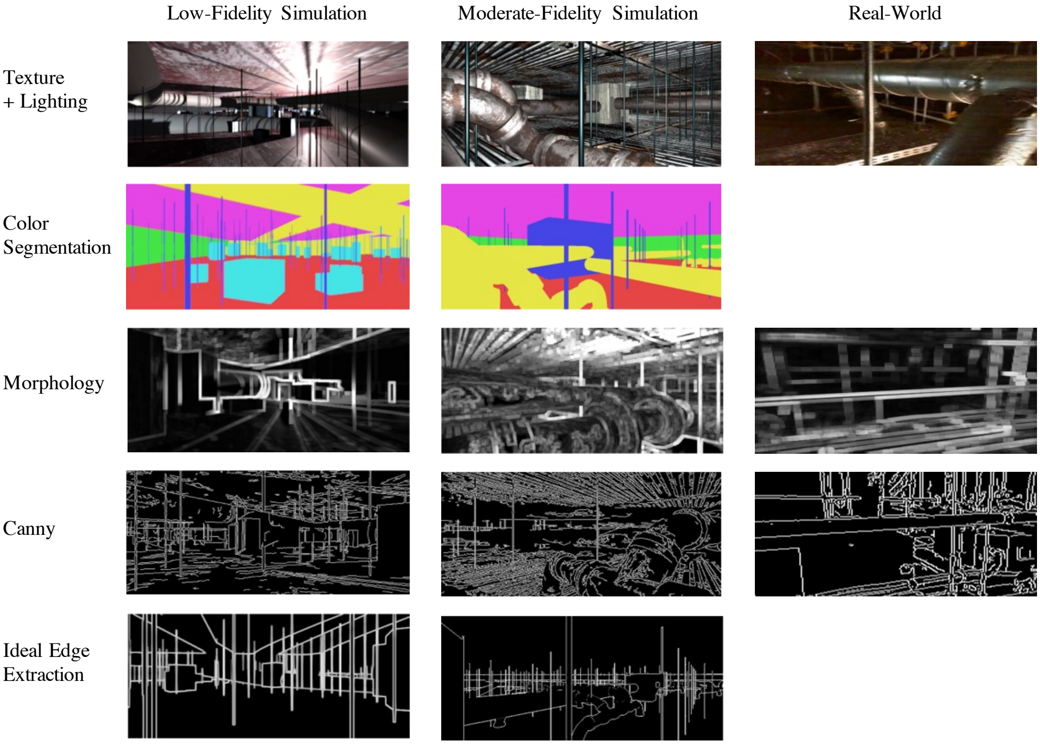


Figure 5. Example images of simulation and real-world data. The original real-world data had a shape for residual neural network (ResNet) input. Color segmentation and ideal edge extraction are the ground truths of the reduced simulation style. In order to obtain these images, specific real-to-sim transfer is required.

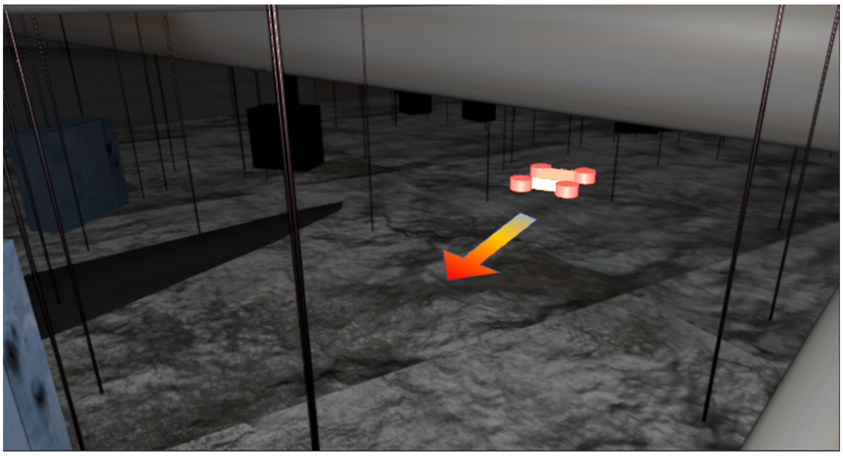


Figure 6. Our virtual MAV had a Rigidbody. When the MAV collided with an object, the front camera saved a rendering image as the screenshot. An edge detection filter, *Robert Cross*, was also implemented in Unity as the post-processing effect. We recorded the original data and post-processed data separately.

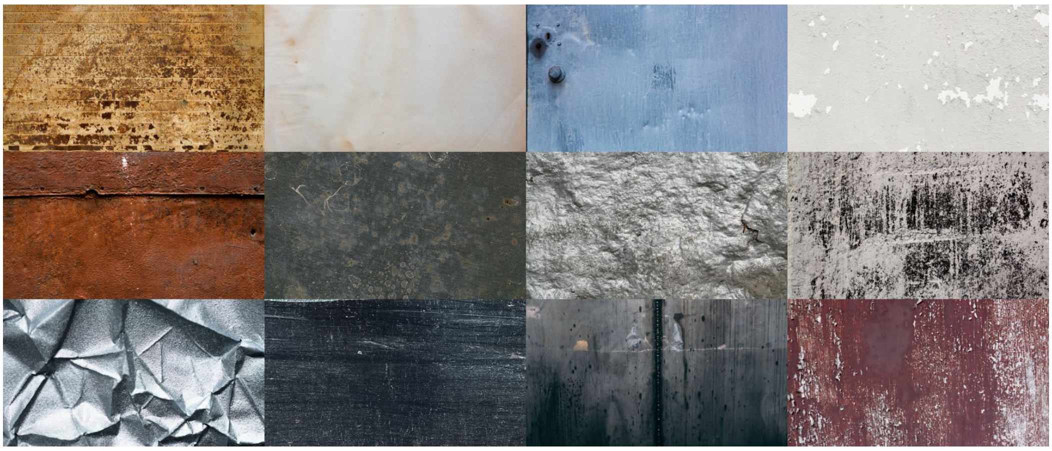


Figure 7. The textures are represented by these images in low-fidelity simulations.

5.1.1. Low-Fidelity Simulation

We prepared three patterns of the texture combination. During image data collection, MAV collided 1500 times. We first create *Texture + Lighting* and *Color Segmentation* from Unity game engine. We applied Canny edge-detection and the morphology method to the *texture + lighting* rendering style to obtain *Canny* images and *Morphology* images, respectively. *Ideal edge extraction* images were derived through Robert Cross post-processing of *Color Segmentation*. This style contained only the outermost edges of the objects.

In this simulation, the position and scale of the objects, including the position, brightness, and color of the light, were randomly determined. However, every object had a simple shape: cylinder or cube. Additionally, each type of object had the same texture. This meant that the simulation had lesser fidelity compared to the conventional sim-to-real methods. Therefore, we defined this simulation as a low-fidelity simulation.

5.1.2. Moderate-Fidelity Simulation

Subsequently, we collected the test dataset to verify adaptation capabilities into realistic simulation data. In order to set up the simulation, we manually created a photorealistic model of the entire environment based on Unity assets. As we mentioned in the introduction, customization of a 3D model of a narrow or confined space has a high cost. Therefore, we adopted and assembled an off-the-shelf piping model and defined this second simulation as the moderate-fidelity simulation. By using this simulation, MAV passed and collided 200 times.

5.1.3. Real-World

Finally, we obtained real-world data, which mimicked the ceiling environment from our experimental site (Figure 8). The sample images obtained from the site are shown in Figure 9. The MAV we used was DJI Tello [31]. We chose it as an off-the-shelf and appropriately sized MAV. The virtual MAV that we selected had a similar field of view and similar dimensions. The real-world dataset consisted of 100 collision and collision-free images. During this phase, we operated the MAV perpendicular to the direction of the camera or in the horizontal direction. MAV collided with the objects at various angles.

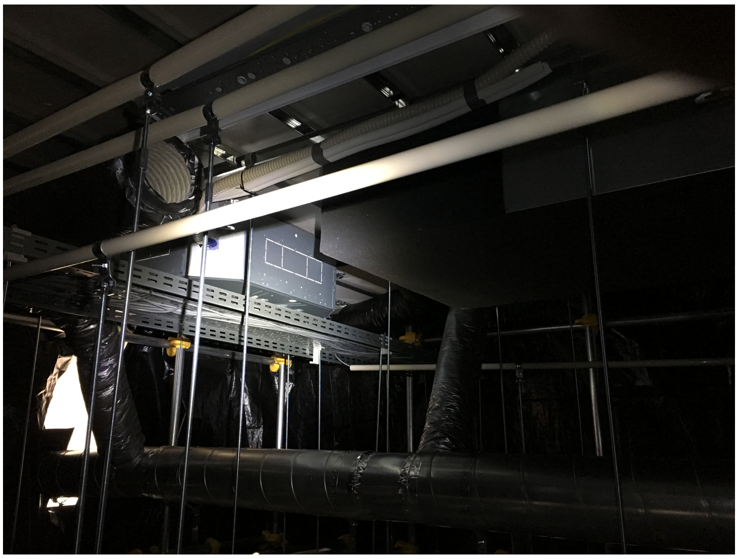


Figure 8. Our experiment site of the ceiling environments. The width and length was 3.5 m, and the height was 1.5 m. This site included some hanging bolts and an air conditioner and a duct pipe.

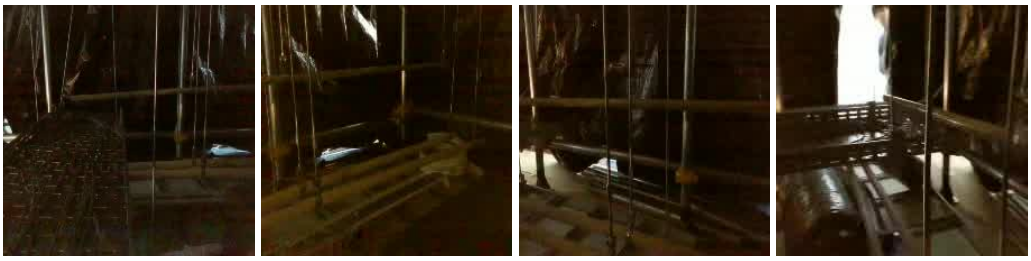


Figure 9. Sample of images obtained from DJI Tello at our experiment site of the ceiling environments.

5.2. Image Based Collision Detection Model

Following conventional research [4], we tested our proposed concept by using binary collision classification by ConvNet. In our study, the network architecture was changed to ResNet because of its similarity to human vision as reported in the literature [32].

We used pretrained *ResNet18* implementation in pytorch with 256 × 256 input size and trained it with 50 epochs, a batch size of 16, and learning rate of 0.01. Our processing PC comprises CPU Ryzen 5 4600H with GeForce GTX 1650Ti GPU.

We trained the models with five rendering styles: *Texture + Lighting*, *color segmented*, *canny*, *morphology*, and *ideal edge extraction*. All models were trained by a subset of the low-fidelity simulation dataset. The models were then evaluated with a test subset of low-fidelity simulation, excluding the subset used for training and for the entirety of moderate-fidelity and real-world datasets. During training, we employed the Adam optimization and cross-entropy loss.

5.3. Experimental Evaluation

Table 3 shows a summary of experimental evaluations. For the test, by a split of the low-fidelity dataset, *ideal edge extraction* outperformed the others. *Color segmentation* produced richer visual features than *ideal edge extraction*. Figure 10 shows the samples of fake-free of *ideal edge extraction* on the moderate-fidelity dataset. These results imply that if the training data does not have sufficient variety of the object's shape, the model does not necessarily require segmentation information. The noisy edge deviated from the texture and decreased the accuracy on *morphology* and *canny*. Texture + lighting also fared worse in terms of achieving adaptation results for the moderate-fidelity simulation.

For the domain adaptation to the real world, *texture + lighting* performed with an accuracy of 0.5. However, the models trained in reduced simulation also exhibited a certain accuracy. Even if the models trained by *ideal edge extraction* received canny data, as indicated in parentheses in Table 3, it performed almost similarly to the canny model. By tuning the hyperparameters, we could improve the performance of the edge-based models.

Table 3. Evaluation in the test subset and adaptation capability. The bold means the best accuracy in each fidelity. The bracket at the lower right means an evaluation using Canny edge extraction images instead of ideal edge extraction of real images.

Rendering Style	Low Fid.	Moderate Fid.	Real World
Texture + Lighting	0.9360	0.8325	0.5050
Color Segmentation	0.9360	**0.8825**	NaN
Morphology	0.9160	0.6525	**0.7200**
Canny	0.8860	0.7850	0.6800
Ideal Edge Extraction	**0.9480**	0.8675	(0.6650)

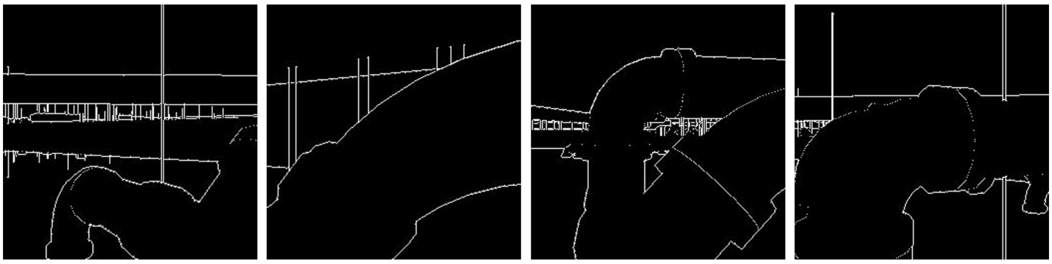


Figure 10. Example of the collision scene, which is mispredicted as the non-collision scene. The characteristic shape of the pipe seems eminent.

5.4. Discussion

In this section, the advantages and limitations of our methodology are discussed. As shown in Table 3, the results of the test datasets in low-fidelity simulation revealed the highest accuracy. Therefore, we thought that the *ideal edge extraction* style had enough visual information to decide if there was a collision or not from a single image. In other words, the edge of an object was the critical information that corresponded to our assumption. Following this point, our proposed pipeline allowed learning flight control policy or method with minimal effort and costs to set up the simulation; this is one of the advantages. If a developer obtains the moderate-fidelity model affordably, diverse shapes are available. When compared to single image-based collision detection, advanced methods, such as reinforcement learning, require a large dataset. In such situations, reducing labor by introducing our concept will be more effective. Finally, regarding data reduction, we can observe that reducing RGB images obtained from simulation to edge detection images yield acceptable results when training based on reduced simulation. Clearly, the amount of image data can be reduced from the three channels of the RGB representation to only one channel of blackwhite or grey-scale image.

A limitation is that our real-to-red.sim concept still requires a robust edge-extraction method and sim-to-real approach in order to operate in real-time. As shown in Figure 5, the morphology style cannot yield a similar image as that acquired through the *ideal edge extraction* style. Canny was not robust to the noisy edge, such as derived from a texture. Thus, we should have pursued a more effective real-to-red.sim transfer. However, as indicated by [24], the separation of advanced visual perception and motion planning is advantageous for domain adaptation. For real-time MAV motion planning, where motion blurring is introduced, candidate applications are limited to non-agile and non-complex trajectories. An appropriate combination of pose estimation and robust control methodology with our learning-based control is required. Alternatively, adding dynamics randomization to methods such as reinforcement learning is also a promising option.

Finally, Our approach enabled MAV to fly in the narrow space environment without creating a map. Although there are many traditional map-based exploration techniques that rely on sensing, mapping and planning, our approach aims for non-map-based exploration. The problem of map-based exploration is that it requires heavy sensors (such as LIDAR for precise measurement) and enormous memory to store map data. Our proposed approach does not require a map for exploration and does not require a heavy sensor. In this study, based on training in reduced simulation, we showed that the MAV can fly in an unknown environment without collision and capture data for post processing of the 3D environment (map). In the future, we believe that we can combine post preprocessing to create a minimal map so that the robot can navigate back to the starting point.

6. Conclusions

In this paper, we proposed the novel real-to-sim concept, *reduced simulation*, to realize autonomous flight in a narrow or confined environment. We also proposed guidelines for creating reduced simulation (texture-lighting less simulation) to cope with the reality gap, which resulted in reducing the development cost. Our user study showed that our proposed reduced simulation pipeline had enough features for visually controlling the MAV. Through the experiments, we confirmed that our methodology possesses an advantage in terms of adaptation capability, without incurring any additional costs compared to traditional cost-saving simulations. Based on the results of these experiments, we believe that reduced simulation is more advantageous than moderate simulations. When compared to sim-to-real approaches, this study is expected to be used in conventional indoor environments, such as in a corridor or in a typical house. Future research can include adding the dilation model to the sequential input or in reinforcement learning. It is also possible to expand input channels or promising reinforcement learning in order to improve the control performance.

Author Contributions: Conceptualization, Y.T., P.R. and T.M.; methodology, Y.T., P.R. and T.M.; software, Y.T.; validation, Y.T.; formal analysis, Y.T.; investigation, Y.T.; resources, T.M.; data curation, Y.T.; writing—original draft preparation, Y.T.; writing—review and editing, P.R. and T.M.; visualization, Y.T.; supervision, P.R. and T.M.; project administration, T.M.; funding acquisition, P.R. and T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partly supported by the KAKENHI Fund for the Promotion of Joint International Research (fostering joint international research (B) No. JP20KK0086) and the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) Grant.

Institutional Review Board Statement: The study was approved by the Ethics Committee of Cybermedica Center, Osaka University.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Faessler, M.; Fontana, F.; Forster, C.; Mueggler, E.; Pizzoli, M.; Scaramuzza, D. Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle: Autonomous, vision-based flight and live dense 3D mapping. *J. Field Robot.* **2016**, *33*, 431–450. [CrossRef]
2. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
3. Xiang, T.; Xia, G.; Zhang, L. Mini-Unmanned Aerial Vehicle-Based Remote Sensing: Techniques, applications, and prospects. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 29–63. [CrossRef]
4. Gandhi, D.; Pinto, L.; Gupta, A. Learning to fly by crashing. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 3948–3955.
5. Loquercio, A.; Maqueda, A.I.; del Blanco, C.R.; Scaramuzza, D. DroNet: Learning to Fly by Driving. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1088–1095. [CrossRef]
6. Kang, K.; Belkhale, S.; Kahn, G.; Abbeel, P.; Levine, S. Generalization through Simulation: Integrating Simulated and Real Data into Deep Reinforcement Learning for Vision-Based Autonomous Flight. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6008–6014.
7. Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Nießner, M.; Savva, M.; Song, S.; Zeng, A.; Zhang, Y. Matterport3D: Learning from RGB-D Data in Indoor Environments 2017. In Proceedings of the International Conference on 3D Vision 2017, Qingdao, China, 10–12 Otober 2017; pp. 667–676.
8. Roberts, M.; Paczan, N. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) 2021, Virtual, 11–17 October 2021; pp. 10912–10922.
9. Höfer, S.; Bekris, K.; Handa, A.; Gamboa, J.C.; Golemo, F.; Mozifian, M.; Atkeson, C.; Fox, D.; Goldberg, K.; Leonard, J.; et al. Perspectives on Sim2Real Transfer for Robotics: A Summary of the R:SS 2020 Workshop. 2020. Available online: http://xxx.lanl.gov/abs/2012.03806 (accessed on 6 December 2021).
10. Xia, F.; Zamir, A.R.; He, Z.; Sax, A.; Malik, J.; Savarese, S. Gibson env: Real-world perception for embodied agents. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
11. Zhang, J.; Tai, L.; Yun, P.; Xiong, Y.; Liu, M.; Boedecker, J.; Burgard, W. VR-Goggles for Robots: Real-to-Sim Domain Adaptation for Visual Control. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1148–1155. [CrossRef]
12. James, S.; Wohlhart, P.; Kalakrishnan, M.; Kalashnikov, D.; Irpan, A.; Ibarz, J.; Levine, S.; Hadsell, R.; Bousmalis, K. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12627–12637.
13. Rao, K.; Harris, C.; Irpan, A.; Levine, S.; Ibarz, J.; Khansari, M. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 14–19 June 2020; pp. 11157–11166.
14. Beyeler, A.; Zufferey, J.C.; Floreano, D. Vision-based control of near-obstacle flight. *Auton. Robot.* **2009**, *27*, 201. [CrossRef]
15. Bills, C.; Chen, J.; Saxena, A. Autonomous MAV flight in indoor environments using single image perspective cues. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 5776–5783.
16. Ross, S.; Melik-Barkhudarov, N.; Shankar, K.S.; Wendel, A.; Dey, D.; Bagnell, J.A.; Hebert, M. Learning monocular reactive UAV control in cluttered natural environments. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 1765–1772.
17. Giusti, A.; Guzzi, J.; Cireşan, D.C.; He, F.; Rodríguez, J.P.; Fontana, F.; Faessler, M.; Forster, C.; Schmidhuber, J.; Caro, G.D.; et al. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. *IEEE Robot. Autom. Lett.* **2016**, *1*, 661–667. [CrossRef]
18. Sadeghi, F.; Levine, S. CAD2RL: Real Single-Image Flight without a Single Real Image. 2016. In Proceedings of the Robotics: Science and Systems 2017, Cambridge, MA, USA, 12–16 July 2017.
19. Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J.J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. The Replica Dataset: A Digital Replica of Indoor Spaces 2019. Available online: http://xxx.lanl.gov/abs/1906.05797 (accessed on 2 August 2021).
20. Armeni, I.; Sax, S.; Zamir, A.R.; Savarese, S. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. 2017. Available online: http://xxx.lanl.gov/abs/1702.01105 (accessed on 2 August 2021).
21. Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; et al. Habitat: A platform for embodied ai research. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 9339–9347.
22. Song, Y.; Naji, S.; Kaufmann, E.; Loquercio, A.; Scaramuzza, D. Flightmare: A Flexible Quadrotor Simulator. 2020. In Proceedings of the Conference on Robot Learning (CoRL) 2020, Virtual, 16–18 November 2020.
23. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*; Springer: Cham, Switzerland, 2018; pp. 621–635.

24. Gordon, D.; Kadian, A.; Parikh, D.; Hoffman, J.; Batra, D. Splitnet: Sim2sim and task2task transfer for embodied visual navigation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 Octobor–2 November 2019; pp. 1022–1031.
25. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
26. Wallach, H.; O'connell, D.N. The kinetic depth effect. *J. Exp. Psychol.* **1953**, *45*, 205–217. [CrossRef] [PubMed]
27. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
28. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 Octobor–2 November 2019.
29. Bradley, J.V. Complete counterbalancing of immediate sequential effects in a Latin square design. *J. Am. Stat. Assoc.* **1958**, *53*, 525–528. [CrossRef]
30. Meyer, J.; Sendobry, A.; Kohlbrecher, S.; Klingauf, U.; von Stryk, O. Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo. In Proceedings of the Simulation, Modeling, and Programming for Autonomous Robots, Tsukuba, Japan, 5–8 November 2012; pp. 400–411.
31. DJI Tello Drone. Available online: https://www.ryzerobotics.com/jp/tello (accessed on 10 July 2021).
32. Wen, H.; Shi, J.; Chen, W.; Liu, Z. Deep Residual Network Predicts Cortical Representation and Organization of Visual Features for Rapid Categorization. *Sci. Rep.* **2018**, *8*, 3752. [CrossRef] [PubMed]