Design and Implementation of Multi-layered Seeing-and-moving-through System

Kosuke Sugisaki
Ritsumeikan University
Shiga, Japan
sugisaki@rm.is.ritsumei.ac.jp

Hideyuki Tamura
Ritsumeikan University
Shiga, Japan

Asako Kimura
Ritsumeikan University
Shiga, Japan

Fumihisa Shibata
Ritsumeikan University
Shiga, Japan

***Abstract*—In this study, we propose a multi-layered seeing-and-moving-through system that allows users to view a multi-layered landscape. In the case where various buildings are present in multi-layer and interrupt users' view, the hidden layer can be viewed using this system. We designed two different modes for the observation: "multi-layered seeing-through mode" in which users can a see-through multi-layered scene without moving around and "moving-through mode" in which users can perceive a feeling of moving around and can directly observe the hidden area. Additionally, we considered two types of presentation method: pre-observed and live. Given that we need to consider the representation methods wherein the system can effectively handle live images, first, we considered a system that uses fixed viewpoint cameras and, then, attempted to utilize drones as movable cameras.**

Keywords—Diminished reality, See-through vision, Drone

I. INTRODUCTION

Diminished reality (DR) is a technique used to visually remove objects by superimposing hidden background images on the objects to be removed [1]. DR technology can be used in various fields. For example, Mann *et al.* used the DR technology to replace an actual object by overlapping it with a virtual object [2]. Rameau *et al.* proposed a system for seeing through a vehicle in front of the driver [3].

The latter DR technique is categorized as "see-through vision" where actual objects are covered with images of their occluded background to render them virtually invisible [1]. Tsuda *et al.* analyzed several methods for a see-through representation, such as wireframe and bird's eye view. They examined the best combination for which the observer can intuitively perceive the space [4].

As one of the DR applications, we proposed a "multi-layered see-through telescope" [5]. The proposed telescope is useful for viewing scenes that are occluded, such as buildings. However, its position and orientation were fixed to ensure that the user can view a limited area only.

Meanwhile, one of the most popular applications for observing an extremely wide area is Google Street View. This technology provides panoramic views from arbitrary positions in many streets worldwide. Users can look around the pre-observed landscapes and change their viewpoints along the streets. However, if the users want to view the rear end of buildings and other obstacles, they need to change their viewpoint using a roundabout route, sometimes resulting in direction confusion. Additionally, users cannot view live images despite they can select previous images.

Considering these backgrounds, we propose a multi-layered seeing-and-moving-through system that allows users to view a multi-layered landscape. In this study, we define "layer" as an area that is separated by obstacles, such as buildings. The proposed system has two different modes: "multi-layered seeing-through" and "moving-through," for viewing a multi-layered scene. By utilizing the "multi-layered seeing-through mode," users can see-through the back of the occluding objects, such as buildings. By employing the "moving-through mode," users can walk around a multi-layered scene and directly view the hidden area, similar to conventional virtual reality (VR) application. Users can switch between these modes whenever they want and freely explore the multi-layered scene.

The proposed system can be utilized in various ways. For example, one of the ways is to keep tracking the person who wanders from place to place in a city scene. Using the proposed system, the user can track the target person without moving. Although the target person does not pass along the road because she/he passes through a building, the user can easily track this person. Another purpose of the system is to understand the geographical features of the place where the user has never been to. Hence, perceiving the geographical features before actually visiting the place is very convenient. Noticeably, similar things can be done with Google Street View. However, the user may lack the sense of direction because Google Street View provides only the movement along streets. Therefore, we believe that intuitively understanding a wide range of geographical features is possible using the proposed system.

In this paper, we describe the design and implementation of the multi-layered seeing-and-moving-through system that provides two different modes for the observation. We examined the basic functions of the two modes and determined the type of representation techniques to be employed.

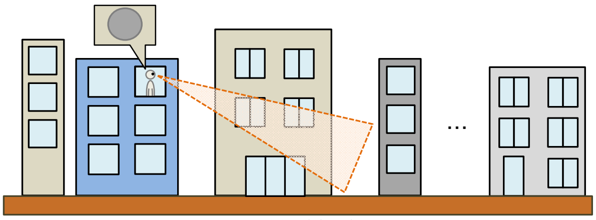
To realize the proposed system, the manner of addressing static information (e.g., geographical map) and live information (e.g., surveillance camera images) should be examined. First, we considered a system that uses fixed viewpoint cameras. We assume that several fixed viewpoint cameras are deployed in a target scene and that geographical map is provided in three-dimensional (3D) CG model. Then, we sought an approach to employ an unmanned aerial vehicle (UAV, commonly known as drone) to capture live images at an arbitrary viewpoint.

II. MULTI-LAYERED SEEING-AND-MOVING-THROUGH SYSTEM

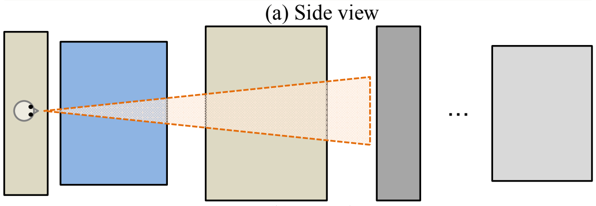
We propose a multi-layered seeing-and-moving-through system that provides two different modes: "multi-layered seeing-through" and "moving-through."

The former mode is based on the concept of a "multi-layered see-through telescope." In this mode, users can view

This work was supported in part by JSPS KAKENHI Grant Number JP17H01747 (Grant-in-Aid for Scientific Research (B)).

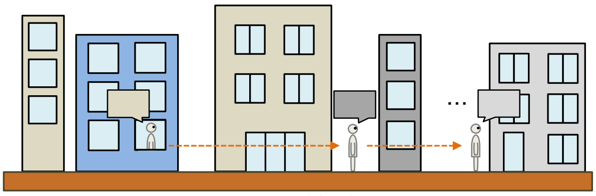


(a) Side view

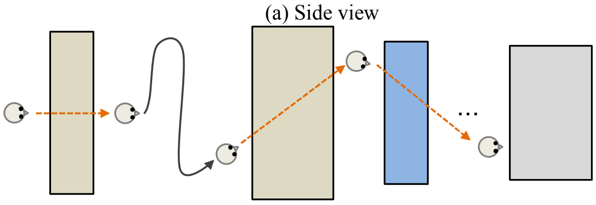


(b) Top view

Fig. 1 Concept of the "multi-layered seeing-through mode"



(a) Side view



(b) Top view

Fig. 2 Concept of the "moving-through mode"

a multi-layered scene without walking around (**Fig. 1**). They can also observe the areas that cannot be seen using the naked eyes due to occlusions, such as buildings. If multiple obstacles are present, a user can observe the hidden area one after another by rendering the target layer existing in the eye direction transparent.

The latter mode is referred to as the "moving-through mode" that enables users to directly explore hidden areas and provides them the feeling of moving around (**Fig. 2**). Unlike the Google Street View that provides only the movement along the streets, this mode allows the users to pass through obstacles.

The goal of this system is to realize a multi-layered seeing-through interface that supports both static information (e.g., geographical map) and live information (e.g., monitoring camera images) in the real world. First, we examined the functions required for the system and implemented a prototype system based on static information, such as textured 3D models whose textures are the pre-observed images of actual scenes (**Fig. 3**).

The functions required for the multi-layered seeing-through mode are as follows:

A. Functions of the multi-layered seeing-through mode

- **See-through window:** Display a part of the hidden layer circularly and vanish the center part of the front layer. Several techniques in viewing a hidden layer are erasing all parts of the front layer, erasing only a part of the front layer, and making the front layer translucent. If the system erases all parts of the front layer, the users will not understand the positional relationship between the front and hidden layer. In contrast, if the system renders the front layer translucent, perceiving the information in the hidden layer is difficult for the users because of the overlapping textures. Therefore, we decided to circularly display a part of the hidden layer and called this circular part as the "see-through window" (**Fig. 4**).
- **Observing layer change:** Change the observing layer forward to the subsequent layer or backward to the previous layer. In this case, the viewpoint (i.e., position of the user) is fixed. Hence, changing the observing layer forward or backward as many times as needed is possible because the aim of the proposed system is to effectively observe the multi-layer.
- **Window shape change:** Resize the see-through window and modify its shape (**Fig. 5**). By changing the shape, as well as its size, we can modify the proportion of the observation area of the hidden and front layer to collect information. Its default shape is

Fig. 3 Image of an actual scene and created 3D CG models from the scene

Fig. 4 See-through window

Fig. 5 Deformed window

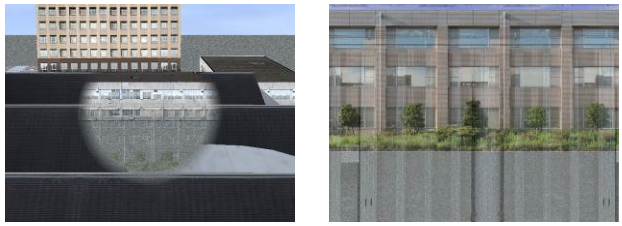


Fig. 6 Different changes between the edge and other pixels in both modes

Fig. 7 Visibility of the first and next layer in the moving-through mode

circle because we obtained the most sense of viewing at the hidden layer.

- **Pan and tilt:** Change the line of sight horizontally and/or vertically. Rotating the field of view enables users to understand the surroundings easily.
- **Zoom in and zoom out:** Expand or shrink the field of view. As the distance to the target hidden layer increases, its visibility decreases. Therefore, we adopted zooming functions to address this issue.
- **Switching between pre-observed and live images:** Switch between pre-observed images and live images. In the case where a user wants to know the geographical features of an unknown place, viewing pre-observed images, such as Google Street View, is sufficient for the user. Meanwhile, if the user wants to watch the current situation of the place that has obstacles, he/she needs switch from pre-observed to live images.
- **Switching between seeing- and moving-through mode:** Switch between the multi-layered seeing-through and moving-through mode anytime.

We made several refinements while implementing a prototype system. The first idea is regarding the boundary of the see-through window. According to the experience of a previous work related to DR [6], we adopted a representation method that blurs the boundary of the see-through window by alpha blending to give the feeling that the hidden layer is visible through the front layer.

The second idea is related to the transition from the front to the hidden layer. If the system instantaneously displays the hidden layer, understanding that the front layer becomes transparent may be difficult for the users because they are observing the hidden layer. Therefore, when the users utilize the see-through window, the blending ratio of the front and hidden layers does not instantly change but gradually. Additionally, the system varies the transition speed of the detected edge and other pixels to show the difference between the front and hidden layer. The reason is when both the front and hidden layers contain many similarly colored pixels, information on the hidden layer and the relative positional relationship are difficult to understand (**Fig. 6**).

The third idea is regarding focus blur. To improve depth sensation, the image of the hidden layer should be blurred at the beginning of the transition to the hidden layer. As the transition continues, the image of the hidden layer is sharpened gradually.

The functions required for the moving-through mode are as follows:

B. Functions of the moving-through mode

- **Move through obstacles:** Move through to the far side of the obstacles in front of the user. Normally, the backward movement is also provided in the same manner. In the moving-through mode, unlike the multi-layered seeing-through mode, one layer is displayed in the entire screen (i.e., all the viewing area) to give a pseudo-movement feeling (**Fig. 7**).
- **Walk around:** Change the viewpoint and viewing direction within the layer. Unlike the multi-layered seeing-through mode, the viewpoint can be freely modified similar to a conventional VR application to explore the observing layer.
- **Return to the initial position:** Stop exploring and return to the initial position.
- The following functions are the same as the multi-layered seeing-through mode: **pan and tilt**, **zoom in and zoom out**, **switch between pre-observed and live images**, and **switch between seeing- and moving-through mode**.

Similar to the multi-layered seeing-through mode, we performed some modifications. The ingenuity related to the transition between layers is adopted in the same manner.

The second idea is regarding the velocity change during a movement. To give users a pseudo-movement feeling, the velocity of the movement is gradually increased while they are moving through to the hidden layer.

III. Showing Live Images in the Move-Through Mode

As our next step, we started to examine how to handle live images in the moving-through mode. When using the proposed system, the pre-observed images would be sufficient in some cases, but sometimes users want to view current live scenes, such as moving cars and walking people. Hence, we introduced live images using fixed viewpoint cameras.

A. Purpose of displaying live images

To let users to know where the fixed viewpoint cameras are mounted, live images are placed in textured 3D models with wireframe pyramid representing the field of view of the camera (**Fig. 8**). The system provides users an interface for identifying the position and direction of the cameras simultaneously. Therefore, they can select the camera that they would like to watch effectively.

The technique of displaying live images in the system is important for the users. If the system only switches live images of the fixed viewpoint cameras and displays them in full screen, then determining where they are currently observing is challenging for them. Hence, understanding which images of the fixed viewpoint cameras are being viewed is necessary when observing live images in all of the field of view.

To address this problem, we introduced the virtual camera motion to show the position and direction of the selected camera. Users can watch the surroundings of the selected camera while a virtual camera moves; therefore, perceiving the situation becomes easy. We used a Bezier curve to draw a trajectory of a virtual camera movement that enters from the rear end (**Fig. 9**). In interpolating discrete points, several interpolation methods, such as B spline curve and Bezier curve, were employed. The reason for using the Bezier curve in this study is to draw a trajectory through the initial point (i.e., current viewpoint) and the end point (i.e., position of the selected camera) (**Fig. 10**).

We also considered the rotation degree of the virtual camera while moving. If the virtual camera rotates at a fixed amount at a time while moving, then the image of the selected fixed viewpoint camera does not appear in the field of view when the direction of the selected camera at the

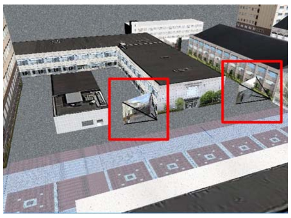


Fig. 8 Live images of fixed viewpoint cameras

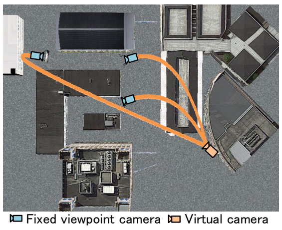
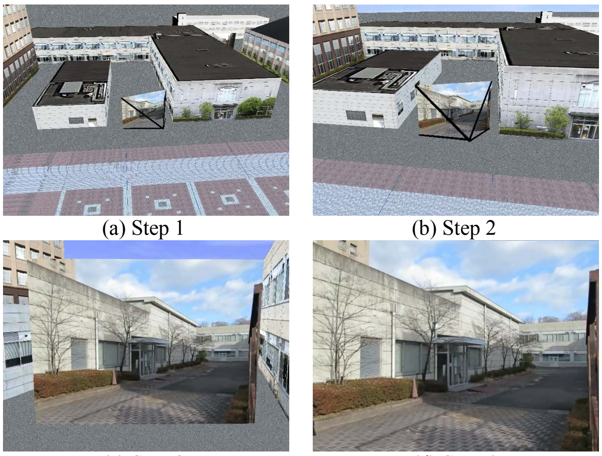


Fig. 9 Trajectory of the virtual camera



(a) Step 1 (b) Step 2

(c) Step 3 (d) Step 4

Fig. 10 Steps of virtual camera motion

initial position is opposite to the direction of the viewpoint. Therefore, the system controls the rotation of the observation field of view to ensure that the image of the fixed viewpoint camera is always displayed at the center of the field of view during movements. This enables the users to always identify the position of the image of the selected fixed viewpoint camera.

Additionally, we compared the moving speed of the virtual camera under the following conditions:

- constant speed;
- speed change, such an inverted parabolic curve; and
- speed change considering the rotation amount.

The second condition denotes that the initial speed is slow, gradually accelerating, and finally decelerating. Under the third condition, the speed of the virtual camera is inversely proportional to the rotation speed of the observation field of view.

B. *Experiment and result*

Experiments were performed to confirm which condition is appropriate for identifying the position of the selected fixed viewpoint camera. We presented three conditions to eight subjects and collected some comments.

Consequently, none of the subjects sensed that the first condition is superior among the three. One subject considered that the second condition is the best, whereas the other seven subjects believed that the third condition is the best. We obtained the following comments:

- Under the third condition, I can understand the position easily because the surroundings of the fixed camera are easy to see compared with other conditions.
- When changing the speed (e.g., an inverted parabolic curve), the feeling of entering the image of the fixed viewpoint camera was strong.
- Under the third condition, being too slow or too fast is not good; therefore, you should find a proper speed.

IV. Introduction of Drone

In our next step, we found a way to utilize drones to capture live images at an arbitrary position. In the previous section, we reviewed the technique of move a virtual camera to display live images. However, blind areas that cannot be observed using any of the fixed viewpoint cameras are unavoidable. To address this problem, we aim to capture live images in vacant places by using drones. For example, Kim *et al.* introduced a real-time video surveillance system using UAV [7].

The first reason why we prefer drones is they can transmit live images via a wireless communication, allowing the drones to move freely.

The second reason is drones can freely move up and down, left and right, and back and forth to ensure that they can move to the place that the users want to view. Although teleoperated robots [8] equipped with a camera can be used as movable cameras, we selected drones because the observed targets are not always on the ground.

The third reason is production of cheap drones becomes high and their use in daily lives is increasing in recently. In the future, a variety of drones would fly around the air. There is a related study to observe a hidden area using a drone by making a hole in the wall in front of a user [9]. The area where live images are captured using the drone is the same, but this research is only limited to one-layer visualization.

A. *Architecture of the system with drones*

To place live images transmitted from drones on textured 3D models, the position and orientation of the drones and live images should be determined. We implemented the proposed system using Unity and decided to operate drones via robot operating system (ROS), a flexible framework for writing robot software [10]. Here, we can control the drones, obtain their position and orientation, capture live images, and check the operations with a simulator by using the ROS.

Figure 11 shows the architecture of the proposed system with drones. In this architecture, the ROS running on Ubuntu and Unity running on Windows are connected through a network. Control commands for the drones are sent to the ROS device driver and then the ROS controls the drones. Positional information and live images captured by the drones are transmitted to the proposed system via a network. By adopting this architecture, the ROS eliminates the difference between virtual drones on the ROS simulator and actual drones to ensure that the proposed system can also ignore the difference.

B. *Preliminary experiment*

We employed Bebop 2 Power Edition as the drone, which is controlled by the ROS, in the proposed system. Bebop 2 is equipped with a custom-made wide-angle lens and an

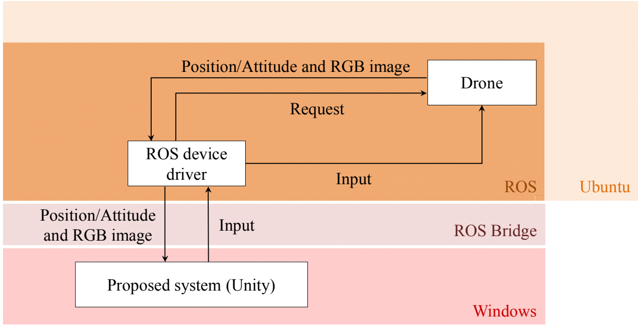
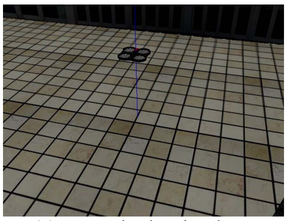


Fig. 11 Architecture of the proposed system with drones

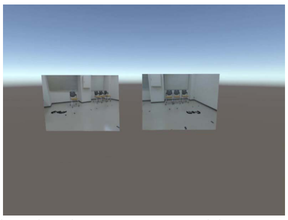


(a) Drone in the simulator

(b) Live image captured by (a) is placed in the textured 3D models

Fig. 12 Result of the simulation

(a) Two actual drones flying side by side



(b) Live images captured by actual drones

Fig. 13 Result of the experiment using actual machines

advanced anti-distortion system to capture top-quality aerial image; hence, it can generate horizontal and vertical 180° images.

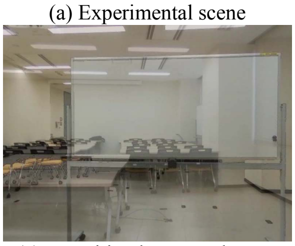
First, we attempted to operate a drone using a simulator (**Fig. 12**). The drone flies in the simulator and transmits live images to Unity via a network. Then, we experimented to control the actual drones in the similar way (**Fig. 13**). Simultaneously, we controlled two actual drones. The drones transmitted live images via a Wi-Fi network. Finally, the system placed the transmitted live images according to the position and orientation of the drones. Additionally, we attempted to realize the transition between two layers in the moving-through mode (**Fig. 14**). The room was separated by a whiteboard, and two drones were hovering at the same time (Fig. 14(a)). The system displayed the transition animation when the user changed the viewpoint from the front to the hidden layer (Figs. 14(b)–14(d)).

C. Discussion

We assumed a situation that drones fly only if necessary. If the drones were always flying, then their battery would be exhausted rapidly. Therefore, we propose that a number of drones should be recharged while on their waiting position, such as on a building rooftop, and the closest drone takes off and flies to the required position when necessary. To realize this situation, the rest is just by combining existing

(a) Experimental scene

(b) Live image of the front layer



(c) Transition between the two layers

(d) Live image of the hidden layer

Fig. 14 Moving-through mode using two drones

technologies because an auto-charging platform for drones has already been published [11].

The proposed system can be specifically used in places where people cannot easily enter, such as a nuclear power plant after a disaster, that have multiple buildings. In this case, the drones can effective in viewing hidden areas.

V. Conclusion and Future Works

We proposed a multi-layered seeing-and-moving-through system that allows users to view a multi-layered landscape. In this system, we designed two different modes: multi-layered seeing-through mode wherein users can see-through multi-layer without movement and moving-through mode wherein users can sense a feeling of movement and observe multi-layer. Then, we considered two types of presentation method: pre-observed and live. The presentation method of the pre-observed images was achieved by developing textured 3D models. However, we should design the technique of combining live images with pre-observed images. First, we examined a system using fixed viewpoint cameras in the moving-through mode and then using drones.

The presentation method for live images is not complete yet, and further examination is required. Moreover, the present live images must be examined in the multi-layered seeing-through mode.

In addition, we will evaluate proposed system. We focused on the design and implementation of the system in this paper, so we have to evaluate the system and measure the user experience of the system in the future. We will compare our proposed system to other VR systems which are similar to our system (e.g., Google Street View).

Currently, we have to specify the takeoff position of the drones in advance. Hence, several position and orientation estimation methods are necessary in the future, for example, high-precision global positioning systems and drone position estimation using Google Street View [12].

Additionally, we must develop ready textured 3D models to use the proposed system because the premise is that several 3D CG models exist. For example, given that 3D structures can be obtained with Google Maps nowadays, if the system employs the data of Google Maps and estimates the drone location, the system could be used anywhere. Normally, the 3D structure data can be gathered using other approaches, for example, LiDARs, which autonomous cars equip.

ACKNOWLEDGMENT

The authors would like to thank Tomoka Ishibashi for her participation in the designing of the system.

REFERENCES

[1] S. Mori, S. Ikeda and H. Saito, "A survey of diminished reality: techniques for visually concealing, and seeing through real object," IPSJ CVA, SpringerOpen, 2017.

[2] S. Mann and J. Fung, "VideoOrbits on eye tap devices for deliberately diminished reality or altering the visual perception of rigid planar patches of a real world scene," Proc. 2nd IEEE ISMR, pp. 48–55, 2001.

[3] F. Rameau, H. Ha, K. Joo, J. Choi, K. Park and I. S. Kweon, "A real-time augmented reality system to see-through cars," IEEE TVCG, Vol. 22, No. 11, pp. 2395–2404, 2016.

[4] T. Tsuda, H. Yamamoto, Y. Kameda and Y. Ohta, "Visualization methods for outdoor see-through vision," Proc. ICAT, pp. 62–69, 2005.

[5] M. Yokota, H. Matsuki, S. Mori, S. Ikeda, F. Shibata, A. Kimura and H. Tamura, "A prototype of a multi-layered see-through telescope using diminished reality," Interaction 2016, pp. 563–564, 2016 (in Japanese).

[6] S. Mori, F. Shibata, A. Kimura and H. Tamura, "Efficient use of textured 3D model for pre-observation-based diminished reality," Proc. IWDR 2015, pp. 32–39, 2015.

[7] H. C. Kim, C. S. Lim, C. S. Lee and J. H. Choi, "Introduction of real-time video surveillance system using UAV," Journal of Communications, Vol. 11, No. 2, pp. 213–220, 2016.

[8] J. Y. Chen, E. C. Haas and M. J. Barnes, "Human performance issues and user interface design for teleoperated robots," IEEE SMC, Vol. 37, No. 6, pp. 1231–1245, 2007.

[9] O. Erat, W. A. Isop, D. Kalkofen and D. Schmalstieg, "Drone-augmented human vision: exocentric control for drones exploring hidden areas," IEEE TVCG, Vol.24, No.4, pp. 1437–1446, 2018.

[10] 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, 2009.

[11] C. Woo, S. Kang, H. Ko, H. Song and J. O. Kwon, "Auto charging platform and algorithms for long-distance flight of drones," IEEE ICCE, 2017.

[12] A. L. Majdik, Y. Albers-Schoenberg, D. Scaramuzza, "MAV urban localization from Google Street View data," IEEE IROS, pp. 3979–3986, 2013.