applied sciences

MDPI

Article

End-To-End Controls Using K-Means Algorithm for 360-Degree Video Control Method on Omnidirectional Camera-Equipped Autonomous Micro Unmanned Aircraft Systems

Jeonghoon Kwak and Yunsick Sung *

Department of Multimedia Engineering, Dongguk University-Seoul, 30 Pildong-ro, 1-gil, Jung-gu, Seoul 04620, Korea; jeonghoon@dongguk.edu

* Correspondence: sung@dongguk.edu; Tel.: +82-2-2260-3338

Received: 1 September 2019; Accepted: 15 October 2019; Published: 18 October 2019

Abstract: Micro unmanned aircraft systems (micro UAS)-related technical research is important because micro UAS has the advantage of being able to perform missions remotely. When an omnidirectional camera is mounted, it captures all surrounding areas of the micro UAS. Normal field of view (NFoV) refers to a view presented as an image to a user in a 360-degree video. The 360-degree video is controlled using an end-to-end controls method to automatically provide the user with NFoVs without the user controlling the 360-degree video. When using the end-to-end controls method that controls 360-degree video, if there are various signals that control the 360-degree video, the training of the deep learning model requires a considerable amount of training data. Therefore, there is a need for a method of autonomously determining the signals to reduce the number of signals for controlling the 360-degree video. This paper proposes a method to autonomously determine the output to be used for end-to-end control-based deep learning model to control 360-degree video for micro UAS controllers. The output of the deep learning model to control 360-degree video is automatically determined using the K-means algorithm. Using a trained deep learning model, the user is presented with NFoVs in a 360-degree video. The proposed method was experimentally verified by providing NFoVs wherein the signals that control the 360-degree video were set by the proposed method and by user definition. The results of training the convolution neural network (CNN) model using the signals to provide NFoVs were compared, and the proposed method provided NFoVs similar to NFoVs of existing user with 24.4% more similarity compared to a user-defined approach.

Keywords: micro unmanned aircraft systems; surveillance; 360-degree videos; deep learning; normal field of view; end-to-end controls

1. Introduction

Micro unmanned aircraft systems (micro UAS) [1] equipped with a camera [2–4] have been used in applications such as traffic surveillance and hobby filming. In addition, data collected from the micro UAS is used to perform real-time monitoring, providing wireless coverage, remote sensing, search and rescue, delivery of goods, precision agriculture, and civil infrastructure inspection for various civil applications [5–7]. Micro UAS has the potential to revolutionize the science, practice, and role of remote sensing in a variety of applications [8]. As micro UAS technology advances, providing various applications for user convenience is possible.

To perform a task, the micro UAS flies autonomously based on the pilot's control or the waypoint set by the pilot [9–11]. During the flight, the camera mounted on the micro UAS captures its surrounding environment. To photograph or track an object for the micro UAS task, it shoots autonomously as

well as through a camera attached to the micro UAS [12–14]. The micro UAS controls its direction to photograph or track the objects, or the camera attached to the micro UAS controls its controllable gimbal in the direction of the objects to be photographed or tracked. When an omnidirectional camera is attached to the micro UAS, the surrounding environment can be photographed without controlling the gimbal.

Using the micro UAS equipped with an omnidirectional camera, the surrounding environment is photographed. The user controls the generated 360-degree video in the direction to be viewed and confirms the objects. The 360-degree video can be automatically controlled based on the objects captured [15–17]; during this process, we require methods to recognize the objects as well as to control the 360-degree video by prioritizing objects. Normal field of view (NFoV) [17] refers to a view presented as an image to a user in a 360-degree video. A problem encountered when tracking one or more objects is that a method is required to provide the NFoVs in a 360-degree video without specifying the objects in it and to allow the user to intuitively view the 360-degree video and specify the NFoVs.

To provide NFoVs, the 360-degree video can be intuitively controlled via the end-to-end control method [17]. The video control signal used herein is the signal that provides NFoV in a 360-degree video. The number of video control signals and video control signals provided to the user is specified by the user, and NFoV is provided using the video control signals inferred based on the images contained in the 360-degree video. If the object you want to observe is on the edge of the NFoV, the screen will change frequently and cause dizziness. There may also be unused video control signals if the user directly sets the video control signals to control 360-degree video. Therefore, there is a need for a method for automatically determining video control signals suitable for 360-degree video.

This paper proposes a method of automatically generating representative video control signals to automatically control 360-degree video for micro UAS controllers. The user views the 360-degree video, maneuvers it in the direction of the objects, and collects the resulting 360-degree video and video control signals. The video control signals to be used as the output of end-to-end control–based deep learning are automatically generated by classifying the collected video control signals through the K-means algorithm [18]. A method for training end-to-end control–based deep learning model is proposed using generated video control signals. Based on the collected 360-degree video and generated video control signals, the deep learning model learns to control the 360-degree video. The trained deep learning model is used to control the 360-degree video, and the user is provided with the NFoVs.

In this approach, NFoVs are automatically provided by the captured 360-degree video; the generated video control signals are set based on the purpose of the video. The proposed method explores possibilities in terms of collecting training data for the user to intuitively train the deep learning model without tagging objects, and the possibility of automatically providing NFoVs using 360-degree video, regardless of flight.

The remainder of this paper is structured as follows: Section 2 introduces related works in this area of study; Section 3 proposes a method for calculating NFoVs for micro UAS controllers using 360-degree videos; Section 4 discusses the experimental results to validate the proposed method; and Section 5 presents the conclusions of this study.

2. Related Works

This section introduces the autonomous flight of micro UAS to track objects and describes a method of controlling a 360-degree video to capture the objects.

2.1. Autonomous Micro UAS for Survaillence

To capture fixed objects, a method to designate the waypoints for the micro UAS flight path is used [9–11]. The object is photographed through a camera mounted on the micro UAS, which flies autonomously based on the waypoints. The pilot sets the waypoints of the micro UAS by considering the locations of the object to be photographed. It is thus possible to shoot objects with a camera attached

to the micro UAS as it flies autonomously with a set flight path; however, if a camera is attached to the micro UAS, it is necessary to set the flight path considering the direction of the camera during flight.

A method that shoots fixed objects by utilizing a three-dimensional (3D) map in which a micro UAS is flying has been developed [19–21]. The 3D map must be configured before the flight of the micro UAS; based on the constructed 3D map, the A* algorithm is used to plan the flight path. However, it is difficult to express moving objects in a 3D map that is constructed in advance. When there is a change in the position of the object to be photographed, the micro UAS may not be able to capture it because of the viewing angle of the camera relative to the object during flight. For this reason, a method of finding the objects to be photographed and controlling the gimbals or micro UAS is needed.

Micro UAS fly around an object and photograph it based on its position [22,23]. Based on the position of objects, particle swarm optimization is used to specify the location where the micro UAS will fly. Alternatively, the micro UAS decides the position to fly based on the position of the object set by the user. Where the micro UAS will fly, the micro UAS will fly in a circular path based on the position of the object and shoot the object with the camera. The micro UAS shoots moving objects based on the set position.

A method for shooting object based on an image previously captured by a micro UAS was previously developed [24–26]. Using the features of the object, it checks whether there is an object in the area being photographed by the camera mounted on the micro UAS. To recognize the object, algorithms such as landmark, deep learning, support vector machine, and particle filter are used to realize if there is an object in the image taken from the micro UAS. If there is an object, the position of the object is calculated by using the position of the micro UAS, the inertial sensor, and the object in the image. The object is captured by tracking the moving object using the change in the position of the object. However, in order to track a target using an object tracking technology, an object's position input or labeling of an object for tracking is required. There is a need for a method for tracking objects without inputting its position or labeling the objects.

When the camera is mounted on the micro UAS to capture the objects to be monitored, a method of controlling the camera is required owing to the limitations with the camera's viewing angle. Because the omnidirectional camera shoots all around it, when it is mounted on the micro UAS, the surroundings of the micro UAS can be captured without the pilot having to consider the camera's orientation during the flight of the micro UAS. However, there is still a need for a method to control 360-degree video from an omnidirectional camera.

2.2. 360-Degree Video Control

A method to track an object included in a 360-degree video taken with an omnidirectional camera and provide users with an NFoV has been previously developed [15,16]. Objects set by the user are extracted from the captured 360-degree video. The object is selected by setting priorities among the objects and NFoV is provided to the user based on the 360-degree video position of the object having the highest priority. When tracking two or more objects, a method of providing an NFoV that considers the relationship between objects is needed.

A saliency map is constructed based on the 360-degree video to determine the NFoV to be provided to the user [17]. Users are provided with the highest score in the saliency map as the NFoV. To calculate the saliency map, the score of each object must be set. If there is a similar score, the change in the NFoV provided to the user may be significant. It is difficult to specify the objects that the user should intuitively observe. When photographing one or more objects, calculating the situation of the surrounding objects and providing them to the user is necessary.

A method has been developed to provide NFoVs by controlling 360-degree videos using end-to-end controls. This provides the user with an NFoV as a video control signal inferred by inputting a 360-degree video. However, if the video control signals for the cube are specified, the boundaries of the NFoVs may not be natural; if the video control signals are specified in detail, unused video control signals may be included; if insufficient video control signals are specified, there may be NFoVs that user cannot

access. Thus, there is a need for a method that automatically calculates video control signals and controls 360-degree video based on end-to-end controls.

3. End-To-End Control-Based 360-Degree Video Control Method Using Video Control Signal Generation Based on K-Means Algorithms

This section describes the process of calculating the NFoVs provided in captured 360-degree videos using end-to-end controls. A method of training a convolution neural network (CNN) model using 360-degree video and generated video control signals is introduced, and a method of providing NFoVs to a user using video control signals inferred using the trained CNN model is described.

3.1. Overview

A method to provide users with NFoVs containing objects in the 360-degree video during the flight of a micro UAS equipped with an omnidirectional camera is shown in Figure 1. In this paper, the 'objects' are those that the user observes using the 360-degree video. A ground control station (GCS) transmits control signals to the omnidirectional camera–equipped micro UAS to make it fly autonomously.

The 360-degree video captured and stored every set time during the ith flight of the micro UAS is defined as Video V_i, with $V_i = [v_{i,1}, v_{i,2}, ..., v_{i,t}, ...]$. The user can watch the Video V_i recorded through the micro UAS's omnidirectional camera during in ith flight. The 360-degree video captured at time t is defined by Image $v_{i,t}$. Image $v_{i,t}$ comprises all surroundings of the micro UAS through a omnidirectional camera mounted on the micro UAS at time t.

The 360-degree video taken from the micro UAS's omnidirectional camera is delivered to the GCS, which calculates video control signals to control the 360-degree videos for the NFoVs with the objects using the received Video V_i. Video Control Signal Set $S_i = [s_{i,1}, s_{i,2}, ..., s_{i,t}, ...]$ is defined as a video control signal for calculating the NFoV of the objects using Video V_i. The video control signal for controlling Image $v_{i,t}$ at time t is defined as Video Control Signal $s_{i,t}$.

The data collection stage allows users to view the 360-degree video and collect video control signals that maneuver it in the direction of the objects. The 360-degree video control data are devised from the collected 360-degree videos and video control signal sets. Preprocessed 360-degree video control data are constructed to be used to train deep learning models, which is accomplished in the model training stage.

In the model training stage, by inferring the 360-degree video input and video control signal onto it, the deep learning model shows improvement over the video control signal of the preprocessed 360-degree video control data.

In the video control stage, the NFoVs are provided to the user using the trained deep learning model. In the GCS, Video V_i and Video Control Signal S_i are delivered to the user. The NFoVs calculated by 360-degree images and video control signals are defined as NFoV N_i, described by $[n_{i,1}, n_{i,2}, \ldots, n_{i,t}, \ldots]$. NFoV $n_{i,t}$ is a view provided to the user as a result of controlling the 360-degree image depending on the video control signal; it is calculated using Video $V_{i,t}$ and Video Control Signal $s_{i,t}$. NFoV $n_{i,t}$ contains objects for observation; users can use either a head-mounted display or a 360-degree video play application to identify NFoVs with objects.

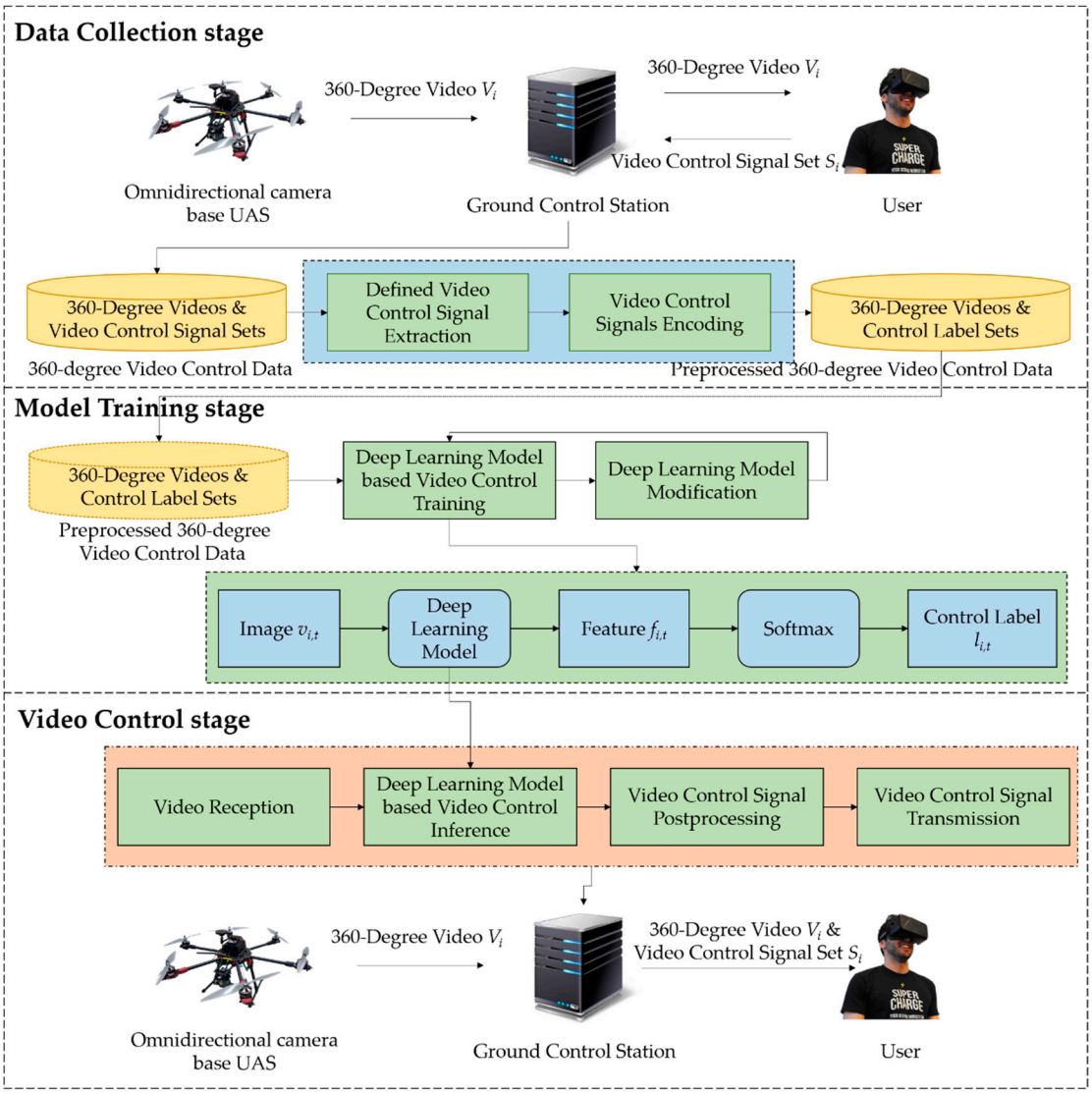


Figure 1. In the process of micro unmanned aircraft systems (micro UAS) autonomous flight, a 360-degree video captured by an omnidirectional camera is used to provide a normal field of view (NFoV) to the user.

3.2. Data Collection Stage

At this stage, a method is used to collect 360-degree video control data based on end-to-end control in order to provide an NFoV of the objects, as shown in Figure 2. When using end-to-end control, the user can intuitively set the NFoV with the objects. The user views the 360-degree video and controls it in the direction of the objects, checks the 360-degree image included in the 360-degree video and checks the NFoV by maneuvering in the direction of the objects. The Video Control Signal $s_{i,t}$ and Image $v_{i,t}$ generated in this process are collected as part of the 360-degree video control data. NFoV $n_{i,t}$ is provided when Image $v_{i,t}$ is controlled by Video Control Signal $s_{i,t}$.

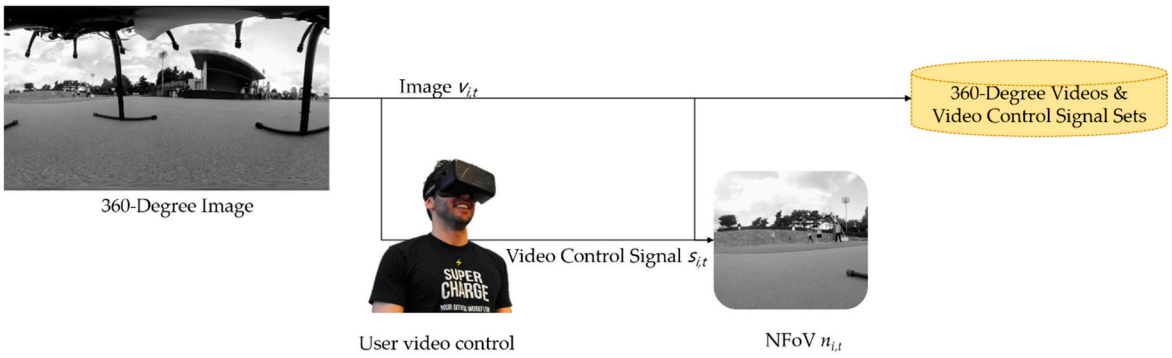


Figure 2. Process of generating 360-degree video control data that provides an NFoV using a 360-degree image and a video control signal.

The preprocessing method using the collected 360-degree video control data is shown in Figure 3. It is a pseudocode for generating preprocessed 360-degree video control data through preprocessing based on 360-degree video control data as shown in Algorithm 1. In the defined video control signal extraction step, representative video control signals are extracted; if all values that can be represented by the video control signals are used, the representative κ video control signals are used to address the increased complexity. The specified κ video control signals are defined as defined video control signals s'_κ, composed of video control signals that, like Video Control Signal $s_{i,t}$, control 360-degree video. The video control signals collected by the K-means algorithm [18] are analyzed to extract the defined video control signals. The κ video control signals are designated to control the NFoVs in which the objects can be placed, and the defined video control signals are set using the results of classification by the K-means algorithm using the collected video control signals, as shown in Equation (1).

$$\arg\min_{S'} \sum_{j=1}^{\kappa} \sum_{S_{i,t} \epsilon S} \|S_{i,t} - S'_j\|^2 \tag{1}$$

where S is $\{S_1, S_2, \ldots S_i, \ldots\}$ and S' is $\{S'_1, S'_2, \ldots, S'_\kappa\}$

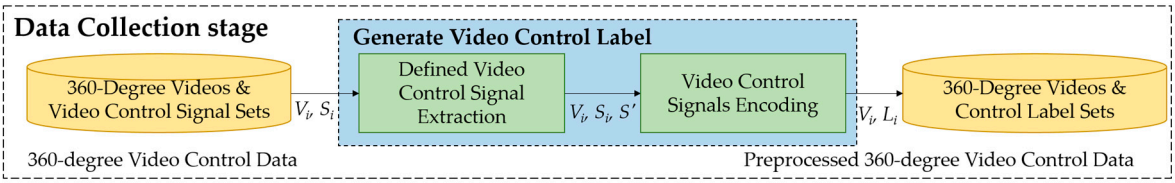


Figure 3. Preprocessing to input 360-degree video control data into the deep learning model.

Video control signals encoding is performed to input video control signals included in 360-degree video control data to a deep learning model using defined video control signals. The labeling of S_i is defined as a Control Label Set L_i, described by $[l_{i,1}, l_{i,2}, ..., l_{i,t}, ...]$. Control Label $l_{i,t}$ is the index of s'_κ. Video Control Signal $s_{i,t}$ finds the nearest defined video control signal with the Euclidean distance and sets the Control Label $l_{i,t}$, as shown in Equation (2). The end-to-end control–based deep learning model is learned using Video V_i and Control Label Set L_i.

$$l_{i,t} = \arg\min_j EuclideanDistance\left(S_{i,t}, S'_j\right) \tag{2}$$

Algorithm 1 Pseudocode for generating preprocessed 360-degree video control data using 360-degree video and 360-degree video control signals.

```
FUNCTION GeneratePreprocessed360-DegreeVideoControlData WITH V, S
  SET κ←CALL Number of video control signal generation
  SET S'←CALL K-means algorithm (S, κ)
  FOR i←1 to S THEN
FOR t←1 to S_i THEN
    SET l_{i,t}←CALL Index of nearest defined video control signal (S_{i,t}, S')
      SET L_i←L_i∪{l_{i,t}}
    END
    SET L←L∪{L_i}
  END
  RETURN V, S
END
```

3.3. Model Training Stage

The process of training to provide the user with the objects based on the collected preprocessed 360-degree video control data is shown in Figure 4. Algorithm 2 is a pseudocode showing the process of learning a deep learning model. In the deep learning model-based control training process, preprocessing is performed so that the collected preprocessed 360-degree video control data can be input to the deep learning model based on videos and video control labels. Image $v_{i,t}$ and the Control Label $l_{i,t}$ are extracted, and the Control Label $l_{i,t}$ is converted into a one-dimensional array by performing one-hot encoding based on the number of defined video control signals. Deep learning model which infer the Control Label $l_{i,t}$ based on Image $v_{i,t}$ is trained. Image $v_{i,t}$ is input to the deep learning model to infer a one-dimensional array of the number of defined video control signals.

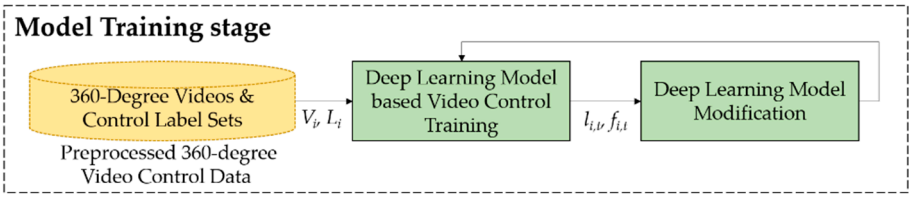


Figure 4. Process of training the deep learning model based on end-to-end control.

A video control signal that controls 360-degree video is inferred through the deep learning model, as shown in Figure 5. Each image is input into the deep learning model as shown in Equation (2), and features are extracted as the output from the deep learning model. Each feature is defined as Feature $f_{i,t}$, and they are represented in a one-dimensional array. The size of the one-dimensional array of features is identical to the number of defined video control signals. In the deep learning model, a process is performed to extract the features of images. Image $v_{i,t}$ is input into the deep learning model, as shown in Equation (3).

$$f_{i,t} = DL(v_{i,t}) \tag{3}$$

where DL denotes the deep learning model.

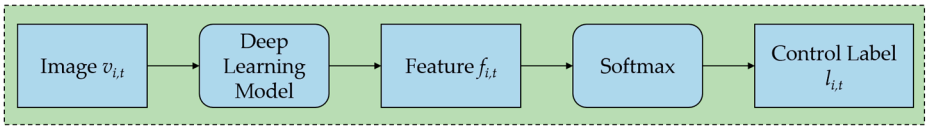


Figure 5. The deep learning model is a process of inferring control signals using video.

The deep learning model is learned by the difference between the Feature $f_{i,t}$ and the Control Label $l_{i,t}$. Feature $f_{i,t}$ is converted into Control Label $l_{i,t}$ through the *softmax* function, as shown in Equation (4).

$$l_{i,t} = Softmax(f_{i,t}) \tag{4}$$

In the deep learning model modification process (depicted in Figure 4), the deep learning model is modified by comparing Feature $f_{i,t}$ and Control Label $l_{i,t}$ inferred from the deep learning model. The correction value is returned to the deep learning model, which is then modified.

Algorithm 2 Pseudocode for training deep learning model using preprocessed 360-degree video control data

```
FUNCTION TrainDeepLearningModel WITH V, L
  FOR i←1 to L THEN
FOR t←1 to L_i THEN
SET l'_{i,t}←CALL Video Control Label Encoding with One-Hot Encoding (l_{i,t})
      SET L'_i←L'_i∪{l'_{i,t}}
    END
    SET L'←L'∪{L'_i}
  END
  FOR ε←1 to Training Count THEN
  SET i←CALL Random extraction
    SET t←CALL Random extraction
    SET f_{i,t}←CALL Inference Deep Learning Model-based Feature (v_{i,t})
  SET value←CALL Compare Feature and Control Label (f_{i,t}, l'_{i,t})
  CALL Modify Deep Learning Model
  END
  RETURN Trained Deep Learning Model
END
```

3.4. Video Control Stage

This stage involves the process of providing an NFoV that shows the user the main objects in the 360-degree video, as shown in Figure 6. In the 'video reception' process, a 360-degree video shot at the micro UAS is input. The 360-degree image input at time t is Image $v_{i,t}$. Algorithm 3 is a pseudocode that generates a video control signal to provide NFoVs by using the received 360-degree video.

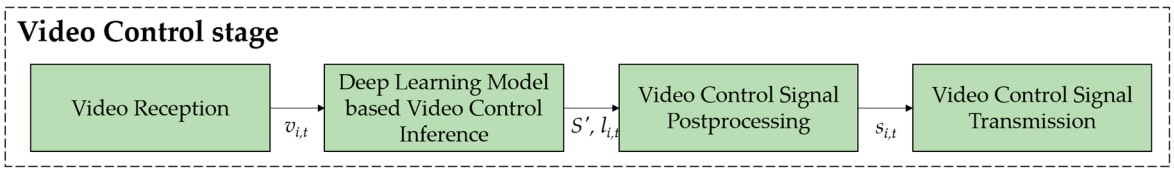


Figure 6. Process of generating NFoV based on a deep learning model that is based on end-to-end control of 360-degree video.

The deep learning model-based video control inference process calculates video control signals based on the 360-degree video; 360-degree images are input to the deep learning model to infer control labels, which are in turn used to generate the video control signals in the video control signal postprocessing process. The Control Label $l_{i,t}$ is converted to the Video Control Signal $s_{i,t}$ as shown in Equation (5). The Video Control Signal $s_{i,t}$ is set to the defined video control signals of the index with the highest value of the Control Label $l_{i,t}$.

$$s_{i,t} = s'_{\arg\max_k l_{i,t}} \tag{5}$$

During video control signal transmission, the calculated video control signal is transmitted to the user; the video is controlled based on the video control signal to output the NFoV containing the objects. An NFoV $n_{i,t}$ is calculated as shown in Equation (6) using an Image $v_{i,t}$ and a Video Control Signal $s_{i,t}$.

$$n_{i,t} = SetNFoV\ (v_{i,t}, s_{i,t}) \tag{6}$$

where $SetNFoV$ denotes a function accepting Image $v_{i,t}$ and Video Control Signal $s_{i,t}$ and predicting the corresponding NFoV $n_{i,t}$.

Algorithm 3 Pseudocode for generating 360-degree video control using trained deep learning model

```
FUNCTION VideoControlSignalGenerationUsingDeepLearningModel WITH V_{i,t}
  SET f_{i,t}←CALL Inference Deep Learning Model-based Feature (v_{i,t})
  SET l_{i,t}←CALL Softmax(f_{i,t})
  SET s_{i,t}←CALL Generate video control signal (l_{i,t}, S')
  RETURN s_{i,t}
END
```

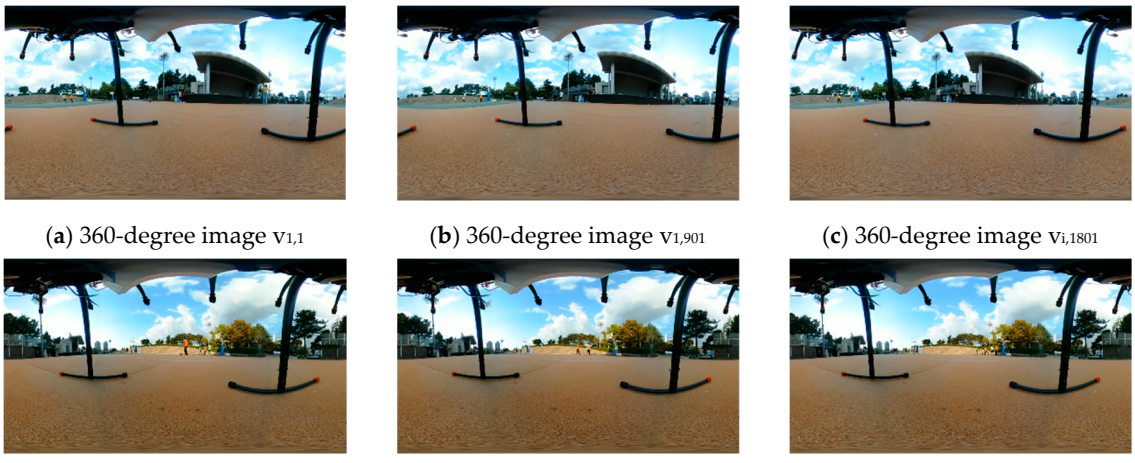
4. Experiments

This section describes the approach used to validate the proposed method for micro UAS controllers. The 360-degree video control data is collected via end-to-end control using 360-degree video. The proposed method and the existing algorithm are compared and analyzed.

4.1. Dataset

In order to evaluate the proposed method, four athletes moving around playing basketball in a school were filmed using an omnidirectional camera, as shown in Figure 7. The 360-degree video, of size 1920 by 1088 pixels, was filmed at 30 frames per second, and each video was collected in about three minutes. A total of 39 videos were collected; 34 of them were used as 360-degree video control data, while the other five were used as evaluation data.

The omnidirectional camera used in the experiment was Ricoh theta Z1 [27]. Two wide-angle lenses can be used to shoot up to 4K (3840 by 1920, 29.97 fps). The omnidirectional camera was equipped with a four-channel microphone for recording sound. It also utilizes its own built-in sensor to provide image stabilization. As the athlete was below the micro UAS, the athletes were attached to the underside of the micro UAS's body to shoot the athletes. Athletes were filmed at five different locations, including cloudy weather, sunny weather, and a little cloudy weather.



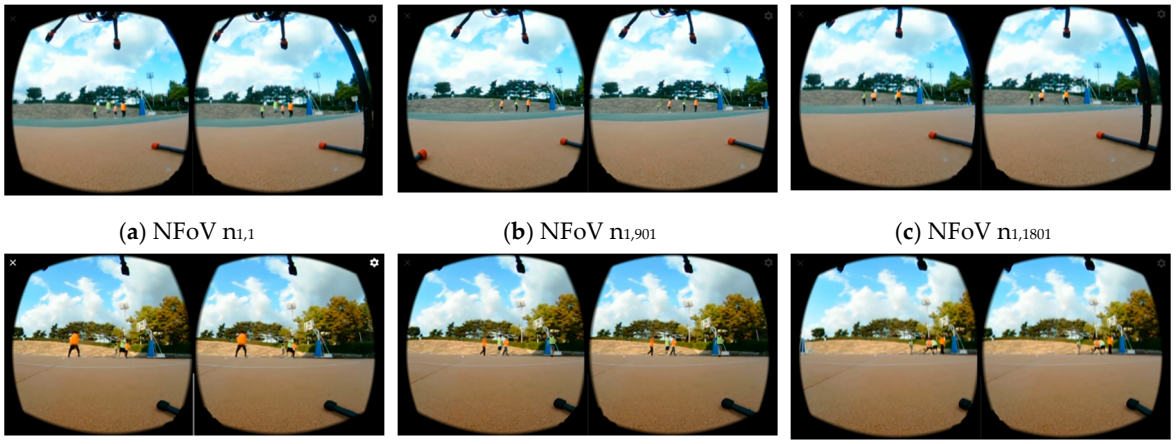
(**a**) 360-degree image $v_{1,1}$ (**b**) 360-degree image $v_{1,901}$ (**c**) 360-degree image $v_{i,1801}$

(**d**) 360-degree image $v_{30,1}$ (**e**) 360-degree image $v_{30,901}$ (**f**) 360-degree image $v_{30,1801}$

Figure 7. 360-degree video of four people playing basketball.

4.2. Process of Generating 360-Degree Video Control Data Based on End-to-End Control

This process involves collecting 360-degree video control data using 360-degree video. Through a smartphone-based application, the user controls the 360-degree video based on the objects in it. The NFoVs are generated by controlling the 360-degree video, as shown in Figure 8; it was photographed to be able to identify the athletes playing basketball. The video control signals were collected differently as the place where the athlete exercised or the position of the athlete changed.



(**a**) NFoV $n_{1,1}$ (**b**) NFoV $n_{1,901}$ (**c**) NFoV $n_{1,1801}$

(**d**) NFoV $n_{30,1}$ (**e**) NFoV $n_{30,901}$ (**f**) NFoV $n_{30,1801}$

Figure 8. NFoVs calculated using the collected 360-degree video and video control signal.

Using the K-means algorithm, κ was assigned to **26** by using the video control signals of the collected 360-degree video control data. The total number of defined video control signals was **26**, and the video control signals consisted of 16 values. The user defined **26** directions to provide all the screens of 360-degree video. Automatically generated video control signals generated by the K-means algorithm were generated similar to the collected video control signals, but in the case of user definitions, 360-degree video was designated for full viewing.

Figure 9 shows the result of converting the collected 360-degree video control data into Control Label L_i. When the first and 30th video control signals were converted into control labels, the K-means algorithm was converted into **12** and **six** control labels, and the user definition was converted into **eight** and **four** control labels. When the athletes were photographed for the first time, the movements of the athletes were higher, and for the 30th, the movements of the athletes were relatively less than that of the first shot. The results of this experiment confirm that the K-means algorithm represents the video control signal well by approximately **1.5 times**.

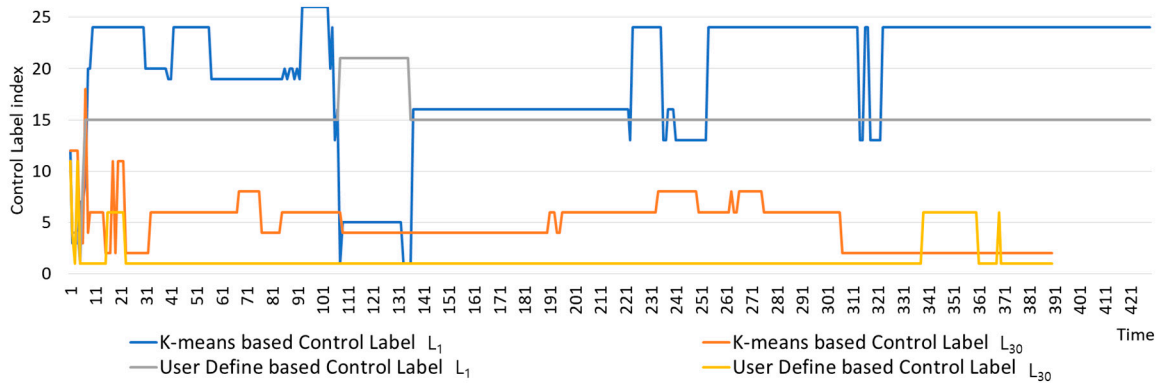


Figure 9. Result of conversion of the first and 30th collected video control signals into control labels.

4.3. CNN Model and Training Result that Generates Video Control Signal to Control 360-Degree Video

The deep learning model used to verify the experiment used a CNN model. Table 1 shows a CNN model for inferring video control signals according to a 360-degree video. The video is converted to a **270 by 480 pixel gray image** and input to the CNN model. Convolutional layers consist of **one dropout, three convolution, and three max pooling layers**. The final result is deduced from the output layer using the results calculated in the convolutional layers.

Table 1. Components of the CNN model.

Layer	Type	Configurations
Input Layer	Input image	270 by 480 grayscale image
Convolutional Layers	Convolution	Number of outputs: 32, kernel size: 1 by 1, strides: 2
	Max Pooling	kernel size: 3 by 3
	Convolution	Number of outputs: 32, kernel size: 3 by 3
	Dropout	-
	Max Pooling	kernel size: 3 by 3
	Convolution	Number of outputs: 32, kernel size: 3 by 3, strides: 2
	Max Pooling	kernel size: 3 by 3
Output Layer	Fully Connected	26 by 1 matrix

Figure 10 shows the change in learning rate during **40,000** epochs using the CNN model. The K-means algorithm–based learning rate converged higher (**to 0.0016**) than the user definition–based learning rate (which converged **to 0.0005**). In the case of training the CNN model based on user definition, the learning rate was lower than the K-means algorithm because the control label did not change much during the shooting of the athletes. The averages of the difference between the converged learning rate and the learning rate from the 10th learning rate (the beginning of the interval where the learning rate does not decrease rapidly) until the end of the learning were compared. The averages of the differences were **0.0005 for the K-means algorithm** and **0.0004 for the user definition**. The averages of the differences were not much different. In addition, it can be confirmed that there were many sections wherein the user-defined learning rate increases rapidly compared to the learning rate of the K-means algorithm.

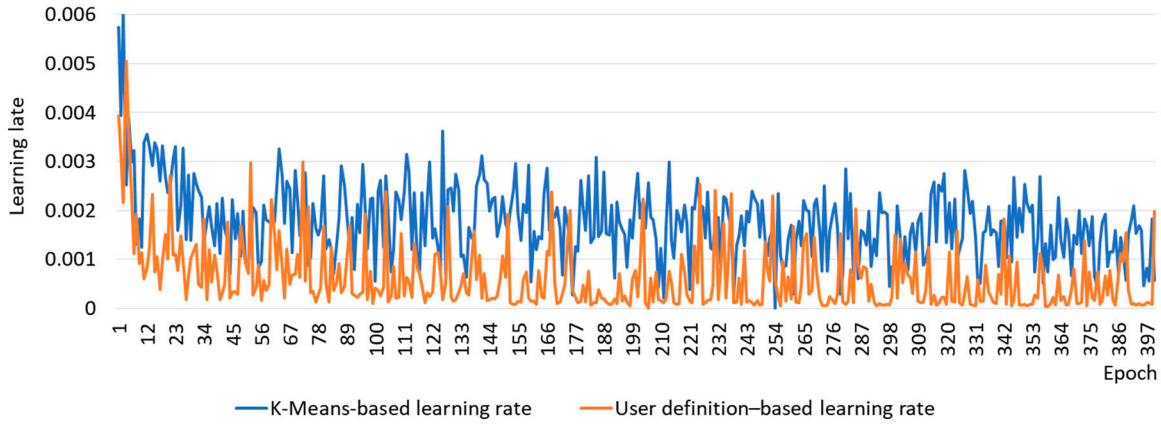


Figure 10. Change in the learning rate when the 360-degree video control signal is learned by the CNN model.

4.4. Experimental Results and Performance Analysis

The results were compared with the collected video control signals using user definition–based as well as K-means algorithm–based defined video control signals, as shown in Figure 11. The difference between the **K-means algorithm**–based defined video control signals and the collected control signal

was **2679.53**, and the difference between the **user-defined** video control signals and the collected control signal was **11,947.28**. The K-means algorithm–based defined video control signals better represent the collected video control signals than the user-defined video control signals. When the video control signal was automatically generated by the K-means algorithm, the difference was consistent with the collected video signals. However, when the control signal is generated based on user definition, the difference occurs differently according to 360-degree video. There was a small section, such as a 360-degree video of 15, 26, among others, or a section with a lot of difference, such as a 360-degree video of two, four, 18, 29, among others.

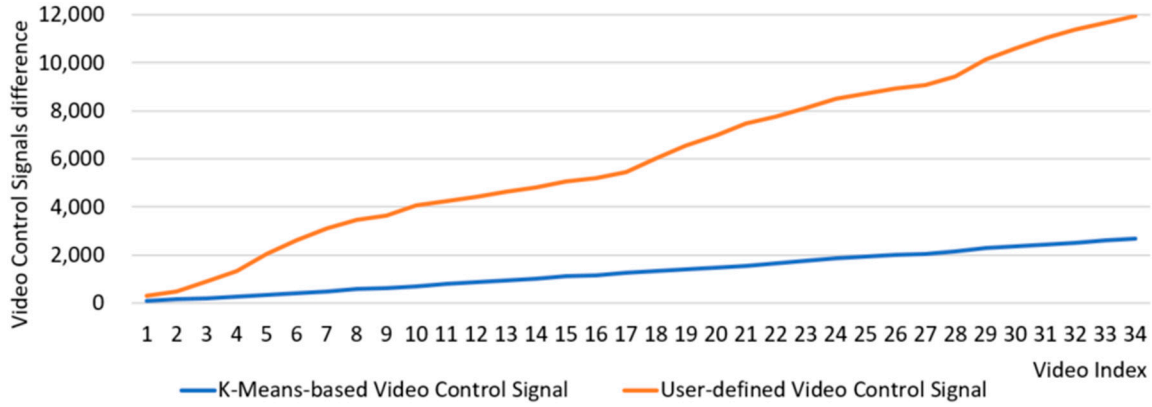


Figure 11. Result of recording the differences between the collected video control signals and the defined video control signals.

The user evaluates the 360-degree video and collects the video control signals. The collected video control signals were compared with the proposed method and the user definition-based video control signals. The proposed method changed the screen **122** times in the 360-degree video, while in the case of user definition, the screen was initially set as shown in Table 2. It was thus verified that the proposed method is more detailed than the user-defined method. The proposed method naturally changes NFoVs according to the athlete's movement. However, although the user definition method has athletes in NFoVs, NFoVs did not change according to the movement of the athletes. For example, if the athletes were on the border within the provided NFoV, the user definition method did not move the NFoV, but the proposed method changed the NFoV so that the athletes were at the center.

Table 2. Number of screen changes in 360-degree video.

Index	Proposed Method	User Definition
1	1	1
2	1	1
3	9	1
4	72	1
5	39	1
Total	**122**	**5**

5. Conclusions

The proposed method generated video control signals for end-to-end control-based deep learning to automatically control 360-degree video. To automatically control the 360-degree video, video control signals for controlling the 360-degree video and the 360-degree video were collected. Using the collected video control signals, video control signals for use in end-to-end control-based deep learning were generated through the K-means algorithm. The deep learning model is trained using the collected 360-degree video and the generated video control signals. The trained deep learning model automatically controls 360-degree video and provides NFoVs to the user.

In the experiment, the training results were analyzed using the CNN model. It was found that based on end-to-end controls, it is possible to generate video control signals that control the 360-degree video, and in the process, the complexity can be reduced by using the K-means algorithm. Using this method, the various NFoVs required by the user could be provided.

Future work entails developing a method to observe the best deep learning model to provide NFoVs to users. There is a need for a method for improving deep learning performance by preprocessing 360-degree video by using object recognition technology and object location technology; such as object tracking. There is a need for a method for increasing the learning rate of a deep learning model by removing unused portions of 360-degree video.

Author Contributions: Conceptualization, J.K. and Y.S.; Methodology, J.K. and Y.S.; Software, J.K.; Validation, J.K.; Writing-Original Draft Preparation, J.K.; Writing-Review & Editing, Y.S.

Funding: This work was supported by the Dongguk University Research Fund of 2017.

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

References

1. Fahlstrom, P.G.; Gleason, T.J. *Introduction to UAV Systems*, 4th ed.; Wiley: Hoboken, NJ, USA, 2012.
2. Wang, Y.; Phelps, T.; Kibaroglu, K.; Sayginer, M.; Ma, Q.; Rebeiz, G.M. 28GHz 5G-based Phased-arrays for UAV Detection and Automotive Traffic-monitoring Radars. In Proceedings of the IEEE/MTT-S International Microwave Symposium-IMS, Philadelphia, PA, USA, 10–15 June 2018.
3. Vahidi, V.; Saberinia, E.; Morris, B.T. OFDM Performance Assessment for Traffic Surveillance in Drone Small Cells. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2869–2878. [CrossRef]
4. Kwak, J.; Sung, Y. Path Generation Method of UAV Autopilots using Max-Min Algorithm. *J. Inf. Process. Syst.* **2018**, *14*, 1457–1463.
5. Sutheerakul, C.; Kronprasert, N.; Kaewmoracharoen, M.; Pichayapan, P. Application of Unmanned Aerial Vehicles to Pedestrian Traffic Monitoring and Management for Shopping Streets. *Transp. Res. Procedia* **2017**, *25*, 1717–1734. [CrossRef]
6. Villa, T.F.; Gonzalez, F.; Miljievic, B.; Ristovski, Z.D.; Morawska, L. An Overview of Small Unmanned Aerial Vehicles for Air Quality Measurements: Present Applications and Future Prospectives. *Sensors* **2016**, *16*, 1072. [CrossRef] [PubMed]
7. 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]
8. Lippitt, C.D.; Zhang, S. The Impact of Small Unmanned Airborne Platforms on Passive Optical Remote Sensing: A Conceptual Perspective. *Int. J. Remote Sens.* **2018**, *39*, 4852–4868. [CrossRef]
9. Santana, L.V.; Brandão, A.S.; Sarcinelli-Filho, M. Outdoor Waypoint Navigation with the AR.Drone Quadrotor. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.
10. Kwak, J.; Sung, Y. Autoencoder-based Candidate Waypoint Generation Method for Autonomous Flight of Multi-unmanned Aerial Vehicles. *Adv. Mech. Eng.* **2019**, *11*, 1687814019856772. [CrossRef]
11. Kwak, J.; Sung, Y. Autonomous UAV Flight Control for GPS-Based Navigation. *IEEE Access* **2018**, *6*, 37947–37955. [CrossRef]
12. Passalis, N.; Tefas, A.; Pitas, I. Efficient Camera Control using 2D Visual Information for Unmanned Aerial Vehicle-based Cinematography. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018.
13. Kwak, J.; Park, J.H.; Sung, Y. Emerging ICT UAV Applications and Services: Design of Surveillance UAVs. *Int. J. Commun. Syst.* **2019**. [CrossRef]
14. Geng, L.; Zhang, Y.F.; Wang, P.F.; Wang, J.J.; Fuh, J.Y.; Teo, S.H. UAV Surveillance Mission Planning with Gimbaled Sensors. In Proceedings of the 11th IEEE International Conference on Control & Automation (ICCA), Taichung, Taiwan, 18–20 June 2014.

15. Hu, H.; Lin, Y.; Liu, M.; Cheng, H.; Chang, Y.; Sun, M. Deep 360 Pilot: Learning a Deep Agent for Piloting through 360° Sports Videos. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
16. Lai, W.; Huang, Y.; Joshi, N.; Buehler, C.; Yang, M.; Kang, S.B. Semantic-Driven Generation of Hyperlapse from 360 Degree Video. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 2610–2621. [CrossRef] [PubMed]
17. Cheng, H.; Chao, C.; Dong, J.; Wen, H.; Liu, T.; Sun, M. Cube Padding for Weakly-Supervised Saliency Prediction in 360 Videos. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
18. Pena, J.M.; Lozano, J.A.; Larranaga, P. An Empirical Comparison of Four Initialization Methods for the K-means Algorithm. *Pattern Recognit. Lett.* **1999**, *20*, 1027–1040. [CrossRef]
19. Yan, F.; Zhuang, Y.; Xiao, J. 3D PRM based Real-time Path Planning for UAV in Complex Environment. In Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 11–14 December 2012.
20. Alejo, D.; Cobano, J.A.; Heredia, G.; Ollero, A. Particle Swarm Optimization for Collision-free 4D Trajectory Planning in Unmanned Aerial Vehicles. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013.
21. Chu, P.M.; Cho, S.; Sim, S. A Fast Ground Segmentation Method for 3D Point Cloud. *J. Inf. Process. Syst.* **2017**, *13*, 491–499.
22. Kim, D.; Hong, S.K. Target Pointing and Circling of a Region of Interest with Quadcopter. *Int. J. Appl. Eng. Res.* **2016**, *11*, 1082–1088.
23. Chu, P.M.; Cho, S.; Park, J.; Fong, S.; Cho, K. Enhanced Ground Segmentation Method for Lidar Point Clouds in Human-centric Autonomous Robot Systems. *Hum. Centric Comput. Inf. Sci.* **2019**, *9*, 17. [CrossRef]
24. Minaeian, S.; Liu, J.; Song, Y. Vision-based target detection and localization via a team of cooperative UAV and UGVs. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 1005–1016. [CrossRef]
25. Park, S.; Jung, D. Vision-Based Tracking of a Ground-Moving Target with UAV. *Int. J. Aeronaut. Space Sci.* **2019**, *20*, 467–482. [CrossRef]
26. Truong, M.T.N.; Kim, S. Parallel Implementation of Color-based Particle Filter for Object Tracking in Embedded Systems. *Hum. Centric Comput. Inf. Sci.* **2017**, *7*, 2. [CrossRef]
27. Ricoh Theta Z1. Available online: https://theta360.com/ko/ (accessed on 26 September 2019).