

# Waypoint and autonomous flying control of an indoor drone for GPS-denied environments

Seung Je Son<sup>1</sup>, Hun Se Kim<sup>1</sup>, Dong Hwan Kim<sup>2</sup>

<sup>1</sup>Department of Mechanical Design and Robot Engineering, Seoul National University of Science and Technology, Seoul, South Korea

<sup>2</sup>Department of Mechanical System Design Engineering, Seoul National University of Science and Technology, Seoul, South Korea

## Article Info

### Article history:

Received Aug 26, 2021

Revised Jul 5, 2022

Accepted Jul 20, 2022

### Keywords:

3D position identification  
Convolutional neural network  
Indoor flying  
Self-location recognition  
Single camera  
Ultra-wide bandwidth module

## ABSTRACT

In this study, we propose a method for recognizing the self-location of a drone flying in an indoor environment and introduce the flying performance using it. DWM1000, which is an ultra-wide band communication module, was used for accurate indoor self-location recognition. The self-localization algorithm constructs a formula using trilateration and finds the solution using the gradient descent method. Using the measured values of the distance between the modules in the room, it is found that the error stays within 10-20 cm when the newly proposed trilateration method is applied. We confirmed that the 3D position information of the drone can be obtained in real-time, and it can be controlled to move to a specific location. We proposed a drone control scheme to enable autonomous flight indoors based on deep learning. In particular, to improve the conventional convolutional neural network (CNN) algorithm that uses images from three video cameras, we designed a distinguished CNN structure with deeper layers and appropriate dropouts to use the input data set provided by only one camera.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Dong Hwan Kim

Department of Mechanical System Design Engineering, Seoul National University of Science and Technology

Seoul, South Korea

Email: dhkim@seoultech.ac.kr

## 1. INTRODUCTION

At present, unmanned aerial vehicles, i.e., drones, have begun to be applied to various fields as they become smaller and more popular [1], [2]. Deloitte, a global consulting firm, estimates that the world's private drones will reach 300,000 units in 2015, with revenues of between 200 and 400 million dollars [2]. The exploration/rescue activities using drones are prominent [3], and they are attracting great interest worldwide. However, until now, civilian drones have been mainly used for broadcasting and hobbies through radio control in outdoor, and automatic flight technology is usually used for global positioning systems (GPS). Chemical engineering science (CES), which was held in Las Vegas in January 2016, was selected as a core technology exhibition to light up 2016 with unmanned vehicles following 2015. The drone is expanding into military business and private business. Recently, Amazon and DHL have been using logistics and delivery systems, and Google and Facebook-led drones have been connecting the global wireless internet connection [3].

British Petroleum diagnoses the damage of Alaskan oil pipeline, and in situ, a subsidiary of Boeing, are studying the migration of glaciers and whales in Alaska through drones. In addition, it applies highly to most industrial and living areas that we imagine, such as solar panel fault diagnosis, emergency patient transportation, weather observation, cleaning and disaster areas, and the applicability is unlimited. Thomas

Fray, an American futurist, said that drones could be used in various disaster disasters in the early warning system and emergency services as 192 drones. Although the development and utilization of drone in the disaster area is active in the private sector in the overseas, the domestic government has started to use the drone at the public level such as forest protection from fire, prevention of citizen safety accident, providing disaster site information [3]. In addition, the flight technology in the room that moves itself to the target position by using the sensors mounted on the drone itself without a GPS is a technology with a high degree of technological difficulty in development and commercialization stages worldwide. Therefore, the participation of research institutes for developing indoor flying is highly needed [4].

Numerous precedent studies on drones have been published. Firefighter developed at the University of California's research team in the US is equipped with an infrared camera and red, green, and blue (RGB) camera to identify the structure of the building as a 3D image, to generate 3D maps and measure the temperature. It is possible to confirm the location of the survivor by securing data such as areas where gas explosions are likely to occur. The firefighting robot Saffir developed at US Naval Research Laboratory is capable of grasping the direction of the fire even in the dust and smoke environment, adopting Lidar sensor. Japanese fire-fighting robots have been developed for firefighting in places where firefighters have difficult access (oil tanks, warehouses) and equipped with waterproof nozzles and hazard identification sensors [5]. A snake-shaped mobile robot (Khoga) has been developed for reconnaissance of collapsed sites such as earthquakes in Japan [5].

It can be controlled wirelessly, and precise reconnaissance in a narrow way is possible through the front camera. The fire-fighting drones developed by the German company Geo-Bon have blown three or four drones into difficult-to-reach firefighters, catch fire points, perform fire-fighting missions, and are equipped with thermal sensors and cameras. A sensor system has been developed by Worcester Polytechnic Institute in USA to allow firefighters to escape before flashover through real-time fire point temperature measurement. W. S. Darley of the United States developed unmanned aircraft system (UAS) and succeeded in scouting the fire scene. The weight is 4.5 pounds, and the flight time is up to 30 minutes. Data measured through the robot can be wirelessly transmitted to the portable terminal of the user.

This work aims to stabilize a drone under a GPS-denied environment such as an office interior environment combined with a proposed position or location estimation technique using ultra-wide band (UWB) modules. A tunnel or an underground parking lot is an environment in which the GPS sensor does not operate, and the structure of the room is invisible due to haze in the event of fire. There have been many works on autonomous navigation of drones that can be used outdoors by camera image and GPS, but some indoor drone flying has been studied in a GPS-free room [6]–[8]. These works use extra sensors such as Inertia module unit (IMU) and are limited to 2D estimation and relative distance estimation, which have limits to direct application to autonomous indoor flying.

As stated before, GPS does not work indoors, and the camera does not work properly either if there is insufficient illumination. Here, it is necessary to develop different methods to recognize the own position of the drone, and control algorithms to conduct safe flight using the identified position information. The current work is to develop a method to recognize the position of drone and an appropriate control system for flying in an indoor environment using the recognized position information. In this work, the control algorithm for stabilization of the drone is introduced, and the hovering flight is performed to evaluate the proposed position recognition.

In this study, a method using CNN structure among artificial intelligence (AI) has been proposed to carry out autonomous flying of drones with the camera images set. This method was implemented by Nvidia in 2017 [9]. However, for better learning results, Nvidia used three cameras to collect and process several data. On the other hand, deep learning-based drone monitoring has been introduced [10], [11], which contributes to expanding the drone applications [10], [12]. Autonomous flying of drones using sensors and a single camera has been found, but autonomous flying with only a single camera and machine learning is not being attempted much. Unlike Nvidia's environment, autonomous flying using a single camera is introduced in this work. The advantage of using a single camera is that it can reduce the computational load on an embedded computer that processes AI algorithms and thus can be implemented with less learning data. To determine if the autonomous flight is possible from the input dataset of a single camera, the simulator was first employed to verify the performance, and the suitable architecture of deep learning was induced and later the effective results were produced through experiments.

## **2. WAY-POINT SYSTEM FOR GPS-DENIED ENVIRONMENT**

### **2.1. UWB communication module production**

To navigate the drone indoors or hover it, the location of the drone should be identified. Without the location information, the drone can only drift. Here, we introduce an indoor localization system for the drone.

First, in order to measure the distance to the drone from the sensor, UWB tag and anchor modules DWM1000 [13]–[15] were employed. Figure 1 shows the designed UWB communication modules. Each module was designed to operate as a tag or anchor by selecting a mode in the program, and it was made compact, which makes it easy to attach to the drone. Arduino microcontroller and DWM1000 module communicate with each other using serial peripheral interface (SPI) communication. The SPI communication protocol is based on the library created by Decawave [15]. The principle of measuring distance between the tag and anchor module using DWM1000 is based on “time of flight method”. In this method, the time taken to transmit and receive communication data is measured and converted into distance using the given formula.

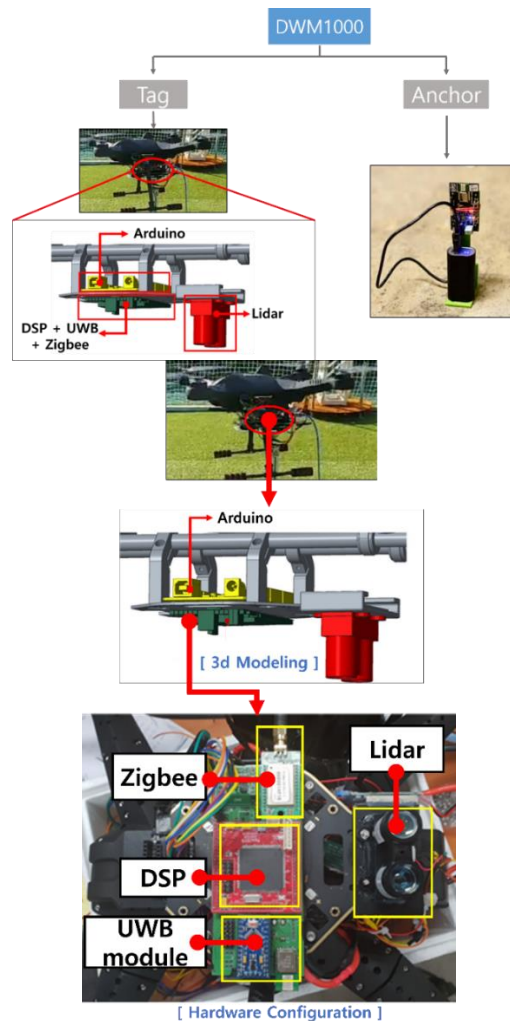


Figure 1. Configuration of DWM1000 for ToF and waypoint module for drone

## 2.2. Way-point system configuration

Attitude control in the roll direction and pitch direction was done using a double proportional-integral-derivative (PID) algorithm. First, the PID control is performed on the roll and pitch angles of the drone [16]–[19]. PID control for each roll and pitch angle is performed once more by using angular velocities in the roll and pitch directions, respectively. The appropriate altitude control algorithm is necessary in order to safely fly indoors. In order to design a robust and stable drone, an altitude controller for the low-flying operation is first designed, and this way employs a P-PD-PID controller.

In order for the drone to fly to a fixed location in an indoor environment where GPS is not recognized, the current location of the drone must be recognized in real time indoors. In this study, using the DWM1000 sensor, which is one of the UWB modules using the time-of-flight (ToF) method, the trilateration method that applies the distance measurement values between multiple anchors and tags installed on the drone is applied to recognize the current position of the drone during flight. After designating various target

positions, it was configured to approach the shortest path from the current drone position to each target position to perform waypoint autonomous flight.

In order to implement the way-point function in the GPS-denied environment, anchors must be installed in a designated space, and the DWM1000 module, which acts as a tag, is configured in drone. Figure 1 shows a system for estimating the position of a drone configured with tag and anchor using DWM1000. In addition, it shows DWM1000 for tag, Arduino mini, digital signal processors (DSP), and Zigbee that process signals in drone. DWM1000 is used to measure the distance between the tag and the anchor installed in the space, and the DSP uses these distance data to calculate the location of the drone using trilateration for numerical analysis.

Arduino enables SPI communication between DWM1000 and DSP. Zigbee is a module for wireless communication with the user to monitor the target position and the current position of the drone in real time. Meanwhile, Anchor is composed of battery, DWM1000, and Arduino mini board.

### 2.3. Communication and signal analysis for way-point module

In this study, the way-point algorithm was designed based on the distance data to the target position and the drone could fly smoothly to the desired target position. Here, the flight controller of the drone used Pixhawk [20], which is the most stable among open platforms. The pulse width modulation (PWM) signal of the used drone controller has a period of 20 ms, and when the control lever is pushed and pulled from the bottom to the top, it can be seen that the PWM high-level signal changes from 1 to 2 ms. By sending this signal to the flight controller by generating PWM through Arduino, the drone can be controlled. In fact, when coded in Arduino, the control value was input as a value between 0 and 1,000 so that the PWM signal was made up to 0 to 100%. Here, if the value is 500, it becomes a neutral value. If it is greater than this value, the drone can move in the forward direction, and if it is less than this value, the drone can move in the reverse direction. According to the difference between this control value and the target position, an appropriate PWM signal is sent to the flight controller to realize the way-point flight of the drone.

Figure 2 shows that when the target position is pointed in the user program written in C#, the distance data to the drone and target is calculated through DSP, and the flight command is issued by sending it to the Arduino. In the image above, the current position of the drone is designated as (0, 0) with a red dot, and the target position is indicated by a green dot, and the user clicks the target point with the mouse to specify it. Dis\_x and Dis\_y are the distances between the target position and the current drone position, and a value corresponding to these values is given to the flight controller. Figure 3 shows the PWM values of roll and pitch according to the flight direction. The flight direction is determined from the Dis\_x and Dis\_y values, and the target position and the drone position can be distinguished according to the negative and positive values. Depending on the positive and negative values, the corresponding PWM values for roll and pitch are designated as large or small values based on 500. This results in a waypoint flight to the target point.

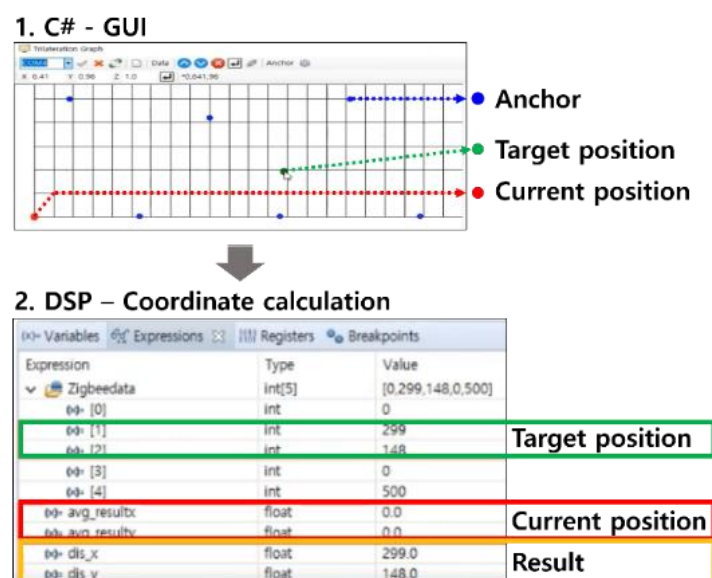


Figure 2. User operating program for drone control

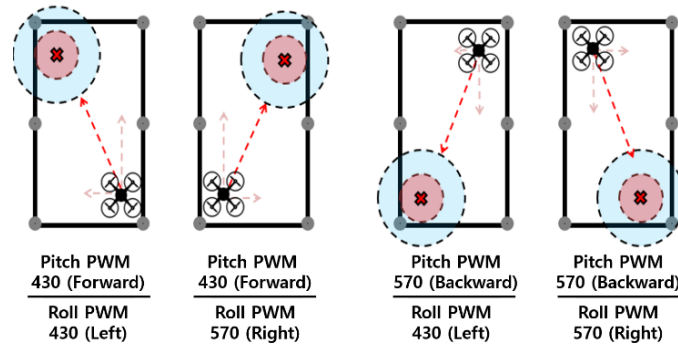


Figure 3. PWM reference values of roll and pitch according to flight direction

### 3. DRONE LOCATION ESTIMATION

The principle of measuring the distance between the tag and anchor module using DWM1000 is based on ToF. In this method, the time taken to transmit and receive communication data is measured and converted into distance.

#### 3.1. 3D location estimation

Figure 4 represents the coordinates of the tag mounted on the drone and the multiple anchors installed in indoor space, respectively. The coordinates of each anchor  $P_i$  are represented by  $(x_i, y_i)$ , and the coordinates of the drone are  $(x, y)$ , which need to be estimated. The distance between the anchor and tag  $r_i$  is measured through UWB sensor, and the estimated distance from each anchor to the drone  $d_i$  is determined as (1). Here, we define error between measured distance and the estimated distance to the drone  $E_i$  in (2).

The process of 3D position estimation is. First, for distance between the  $i$ -th designated anchor and the tag is expressed by (1), including  $z$ -axis data.

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \quad (1)$$

The error between the measured distance and the estimated distance to the drone is also expressed as (2).

$$E_i = d_i - r_i \quad (2)$$

Now, the cost function is defined as (3).

$$J = \frac{1}{n} \sum_{i=1}^n E_i^2 \quad (n \geq 3) \quad (3)$$

In (3),  $n$  is the total number of anchors, and more than three are needed to get a more accurate intersection. If the cost function  $J$  is zero, then three more circles with a radius of  $d_i$ , respectively, form an intersection point completely, which yields in the true location of drone itself, as shown in Figure 4.

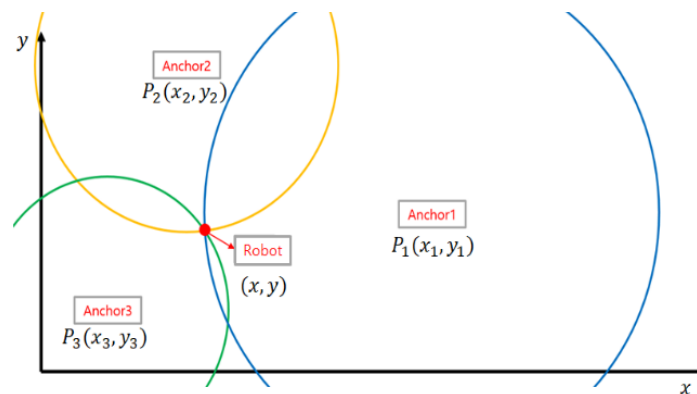


Figure 4. Intersection formed by three measured distances to drone (robot) from each anchor

The formula of the gradient decent algorithm [21] for 3D position estimation is shown in (11).

$$X_{k+1} = X_k - \gamma \nabla J(X_k), X_k = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} \quad (4)$$

Here,  $\gamma$  is the decay rate, and  $\nabla J(X_k)$  is determined as

$$\frac{\partial J}{\partial x} = \frac{2}{n} \sum_{i=1}^n \frac{(x_i - x)(d_i - r_i)}{d_i} \quad (5)$$

$$\frac{\partial J}{\partial y} = \frac{2}{n} \sum_{i=1}^n \frac{(y_i - y)(d_i - r_i)}{d_i} \quad (6)$$

$$\frac{\partial J}{\partial z} = \frac{2}{n} \sum_{i=1}^n \frac{(z_i - z)(d_i - r_i)}{d_i} \quad (7)$$

The update law for the 3D estimation is determined as

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} - \gamma \begin{bmatrix} \frac{2}{n} \sum_{i=1}^n \frac{(x_i - x_k)(d_i - r_i)}{d_i} \\ \frac{2}{n} \sum_{i=1}^n \frac{(y_i - y_k)(d_i - r_i)}{d_i} \\ \frac{2}{n} \sum_{i=1}^n \frac{(z_i - z_k)(d_i - r_i)}{d_i} \end{bmatrix} \quad (8)$$

Using (8), MATLAB was again used to test the performance of estimating the 3D location. The true location was set as (2,7,3), where all units are meters. Figure 5 shows the result of position estimation by testing 100 times. As seen from the result, the average errors for x and y axes are 0.09 m, and 0.28 m, respectively. On the contrary, the error for z-axis is around 1.2 m, which brings about undesirable results. In the 3D estimation using gradient decent algorithm, it is proven that multiple local minimums were found in the process of convergence, which restricts finding the true value. Even if there exists a relatively large error in z-axis, the x and y-axis errors remain close to the result of the 2D estimation. To solve the large estimation error problem at z-axis, an alternative where z-axis value is measured from the external sensor is suggested.

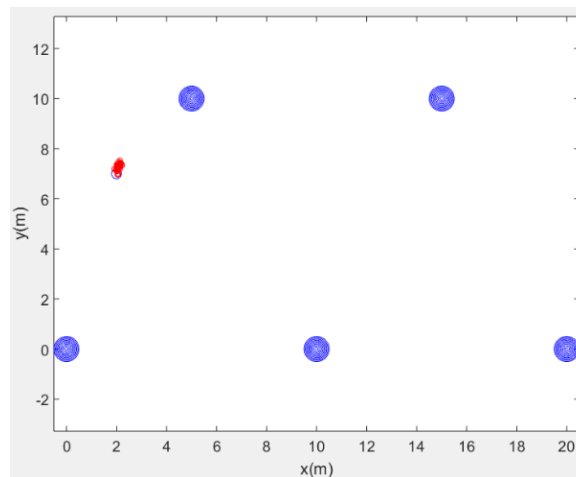


Figure 5. Simulation result of 3D location estimation for the true value of (2, 7, 3)

### 3.2. 2D projection algorithm

Drones flying indoors usually use altitude value by ultrasonic sensors or laser sensors. Rather than estimating all 3D estimation, altitude can be determined prior to solving the 3D estimation, which is based on the gradient decent algorithm. Once the z-axis value is determined by the sensor, we set a projection plane amount to the altitude from the floor, which is depicted in Figure 6.

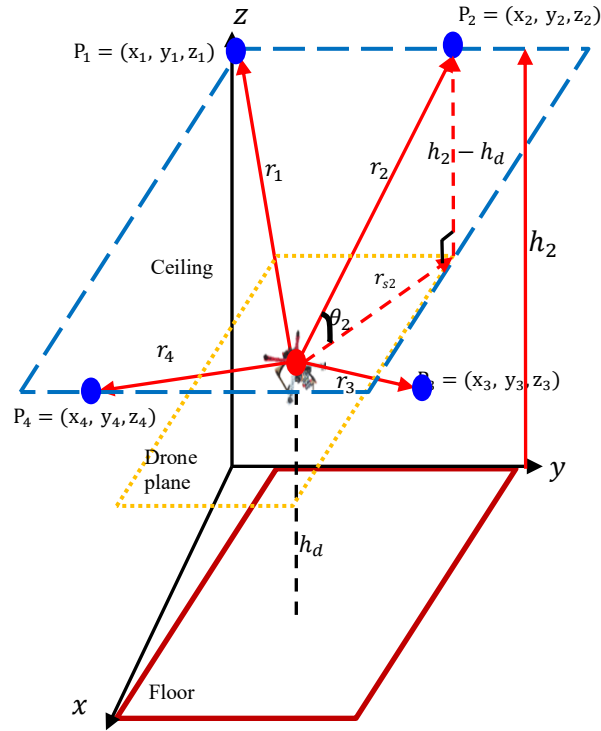


Figure 6. A 2D projection plane generation from distance data from each anchor

Every distance from each anchor is projected to this 2D plane, which varies with respect to the height of the flying drone. Let the altitude of the drone be  $h_d(\cdot)$ , which is measured from the sensor. The measured distance from anchor  $r_i$  is now projected to the 2D plane, which is formed above  $h_d$  from the floor, which is represented as  $r_{si}$ .

$$r_{si} = r_i \cos(\theta_i) = \sqrt{r_i^2 - (h_2 - h_d)^2}, i = 1, \dots, n \quad (9)$$

With this converted value  $r_{si}$ , the estimation update law is determined as

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \gamma \begin{bmatrix} \frac{2}{n} \sum_{i=1}^n \frac{(x_i - x_k)(d_i - r_{si})}{d_i} \\ \frac{2}{n} \sum_{i=1}^n \frac{(y_i - y_k)(d_i - r_{si})}{d_i} \end{bmatrix} \quad (10)$$

With this new estimation law, we tested again 100 times using MATLAB, yielding to average errors of 0.07 m in z-axis, 0.07 m x-axis, and 0.11 m in y-axis, respectively, which is much smaller than the direct 3D estimation result. Also, the experiments were conducted to verify the 2D projection algorithm for 3D position estimation by placing 4 anchors on the wall and one tag on the flying object. The test results are similar to MATLAB simulation results, yielding to 0.07 m in z-axis, 0.09 m x-axis, and 0.12 m in y-axis, respectively. Through these experimental results, we found the proposed scheme is very convincing to apply for indoor 3D position estimation.

To verify the validity of the position values estimated from the algorithm, we used a motion capture device (V120 DUO and Trio from OptiTrack company), as shown in Figure 7, where four DWB100 modules are attached to the wall and the motion capture device stands at the side of the corridor. While the drone is flying in the air the exact 3D position values (here, 2D values are first considered) the motion capture measured the distance. Figure 8 shows the comparisons between the proposed algorithm using UWB1000 modules, motion capture device, and true values. The measured values from the motion capture device were compared with the position estimated by using the proposed algorithm, resulting in close agreement between the two results, which guarantee the proposed position estimation method is reliable.

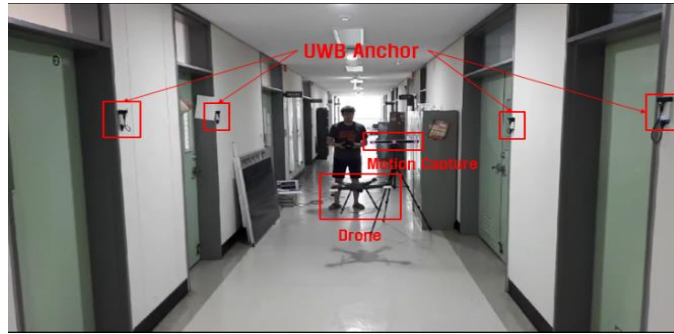


Figure 7. Position verification of flying object in air using a motion capture device

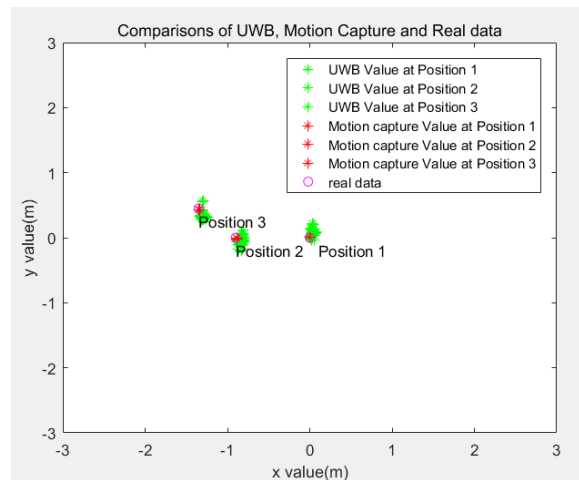


Figure 8. Comparisons between proposed estimation, motion capture device, and true values

#### 4. WAY-POINT CONTROL

Since the drone is an unmanned aerial vehicle, even if it reaches the target position after flying, it does not stop due to inertia and moves further in the direction it was moved. Since the drone is in the air, friction is low and the inertia force is relatively large, so it will fly further in the direction it is traveling. Therefore, different flight strategies were designed for each flight section to improve the accuracy of way-point flight and to solve the inertia problem. That is, a method for offsetting the inertia force with the thrust of the drone was applied by setting the PWM differently for each section. Figure 9 shows the process of setting different PWM values for each flight section. It did not simply fly toward the target position. As shown in Figure 9, the inertia force of the drone was weakened using the counter thrust in section A, and the drone smoothly approached the target position from section B.

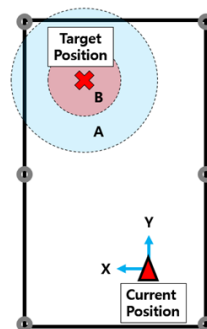


Figure 9. Waypoint PWM by interval

Figure 10 shows the flowchart for performing control to reach the target position. Through this algorithm, we implemented a way-point flight of a drone indoors using the UWB modules without GPS. The results of this experiment will be described in detail later.

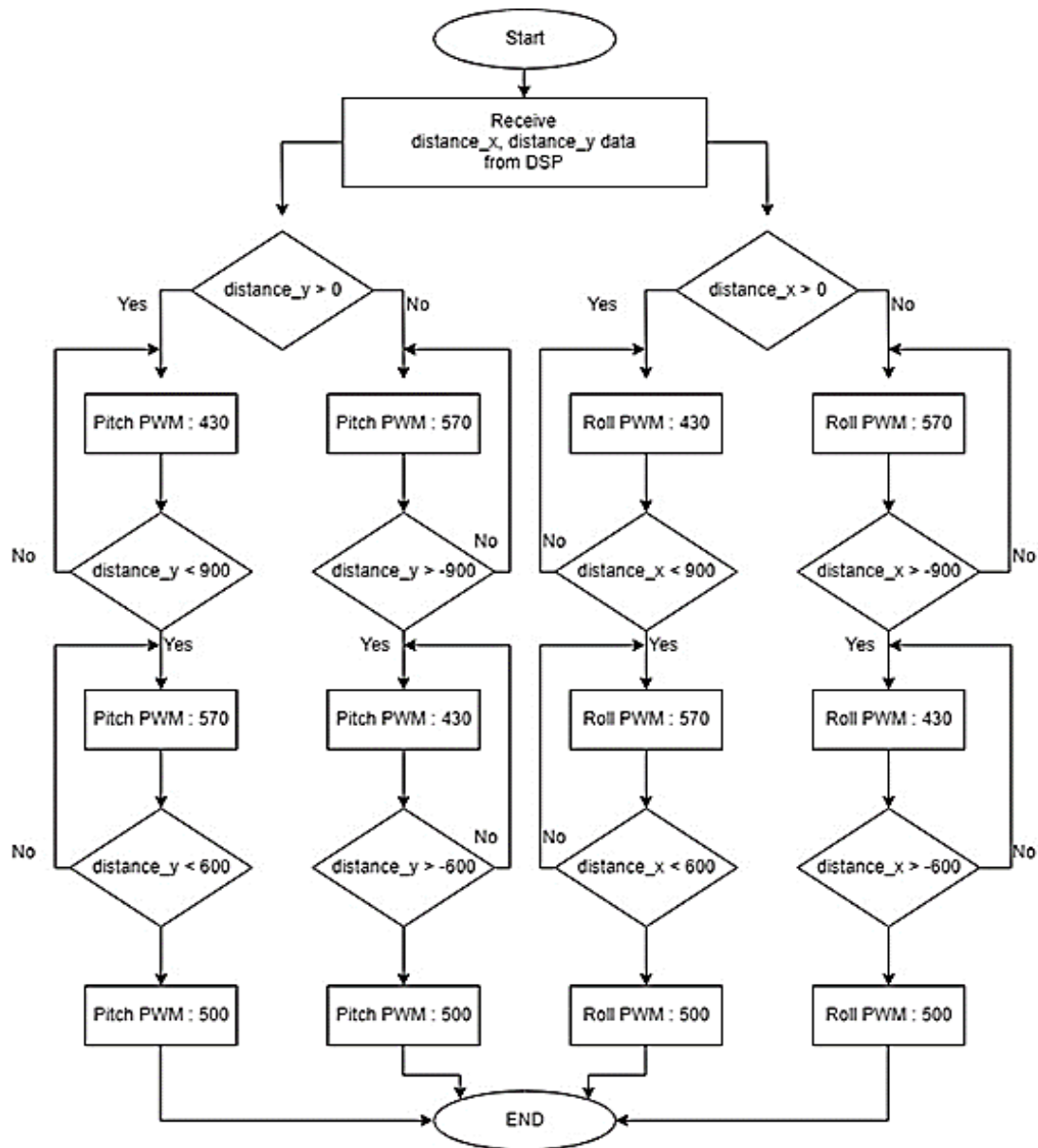


Figure 10. Final algorithm of waypoint

#### 4.1. Way-point control in a near field

After verifying the hovering motion of the drone, a way-point control that is autonomous flying is performed. Figures 11 and 12 show that the drone is controlled through the proposed controller to move from the target value to the designated position and reach the target value again. Figure 12(a) shows the flying of the drone to the target point 1 m behind the start point. Figure 12(b) shows the hovering at the rear 1 m. Next, Figure 12(c) shows a process of returning to the target point and finally landing on the floor.

Figure 13 represents the 3D position while the drone flies and Figure 14 shows the 3D target position command data for x, y, and z directions. Particularly, for a simplicity, target  $z=0$  is set as the actual altitude of 1.5 m. As seen from the results, the drone reaches the target value relatively successfully although there is some fluctuation during flight, which states that the proposed controller and 3D position estimation algorithms are working effectively.

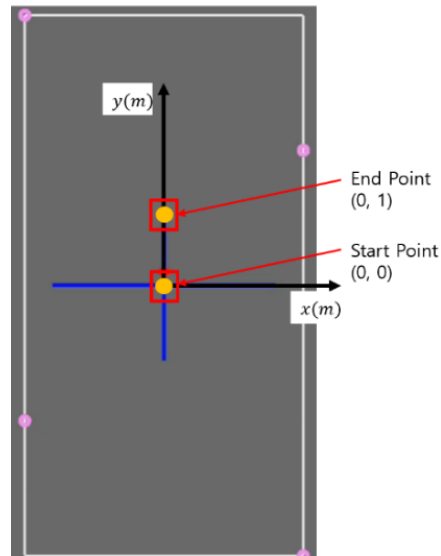


Figure 11. Target and start positions of the drone reaching the target

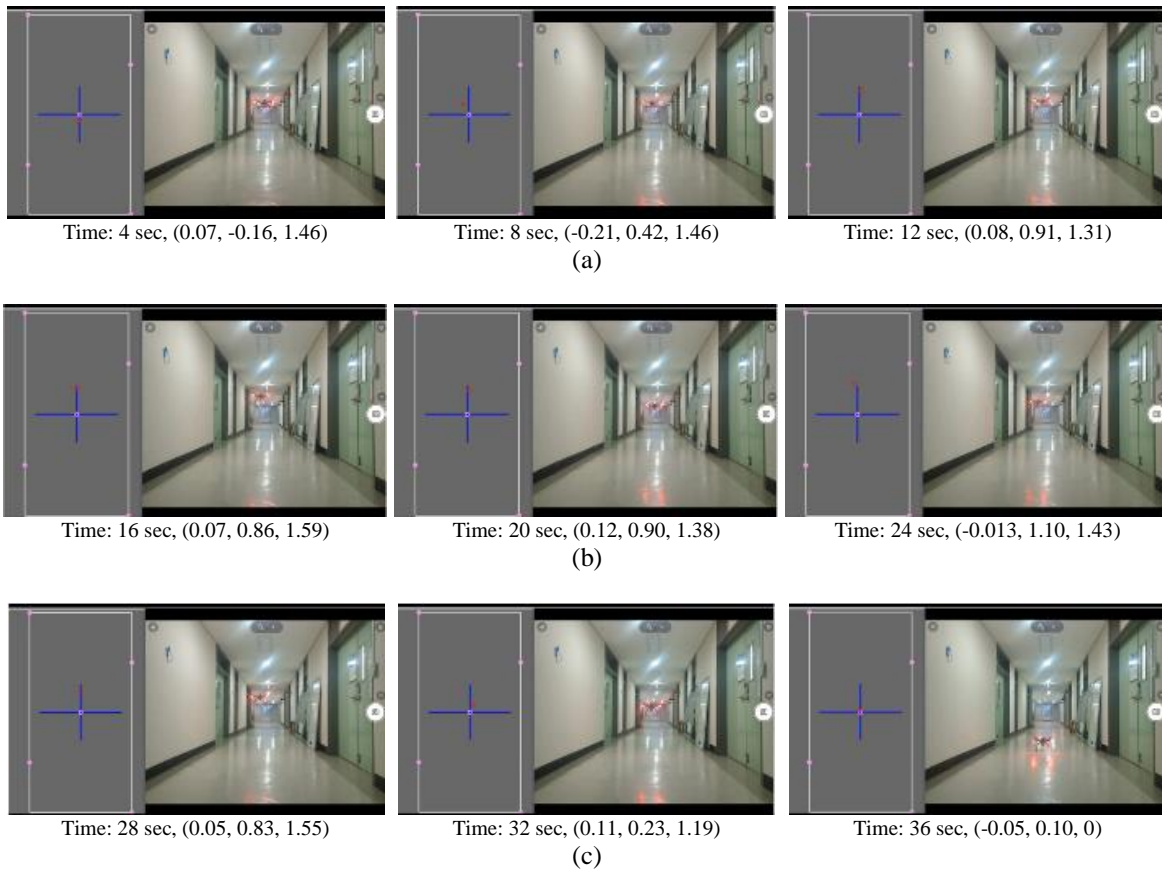


Figure 12. Frame images of demonstration of position control of the drone reaching the target: (a) target point 1 m behind the start point, (b) hovering at the rear 1 m, and (c) returning to the target point

#### 4.2. Waypoint flying in a far field

Figure 14 shows the waypoint flight toward the destination after the drone starts flying in the center of the indoor soccer hall, which is 40 m in length and 20 m in width, respectively, when the target position is specified in the bottom image. The way-point target 1 is set at the middle of the left side of the hall. Another

way-point target 2 is assigned to the middle of the right side of the hall, where it is marked at the bottom image of Figure 15.

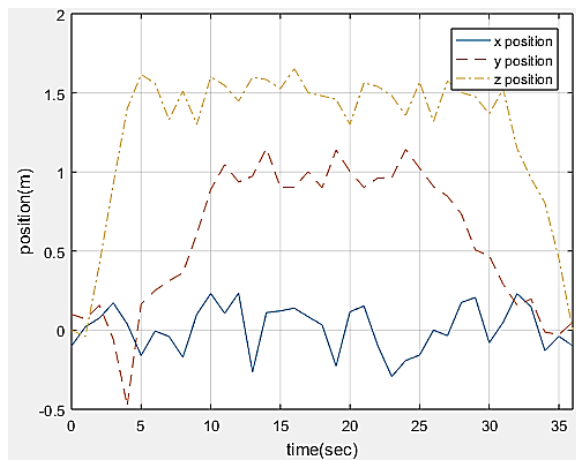


Figure 13. Time response of 3D displacements of the drone while flying with control

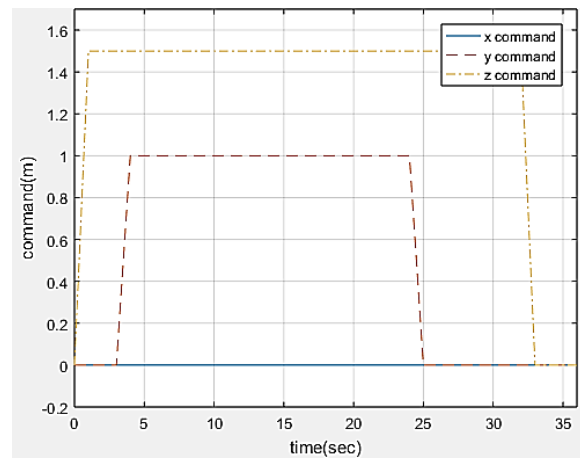


Figure 14. The target 3D position values of drone for flying control (z-target 0 is equivalent to actual altitude of 1.5 m)

By solving the problem of inertia described above, the result of stable flight to the target position was obtained. Here, the output of the motor and pitch and roll angles of the drone determines the speed of the drone. Since the pitch and roll angles are fixed during flight, the speed is relatively constant. In addition, since the altitude control function was added by attaching a lidar, drone was able to fly stably.

Through Zigbee communication, real-time communication between the C# program and the control unit of the drone was enabled. On the top of Figure 15, we can see the map created through the C# program. Also, anchors are marked on the map at a fixed location, and waypoints are possible within the range of anchors. Since the flight commands were set differently for each section, it was possible to confirm that the drone flew stably. Figure 16 shows the drone waypoint flying to the target position indicated in the top image. Fifty round trip flights were conducted to ensure safe waypoint flying.

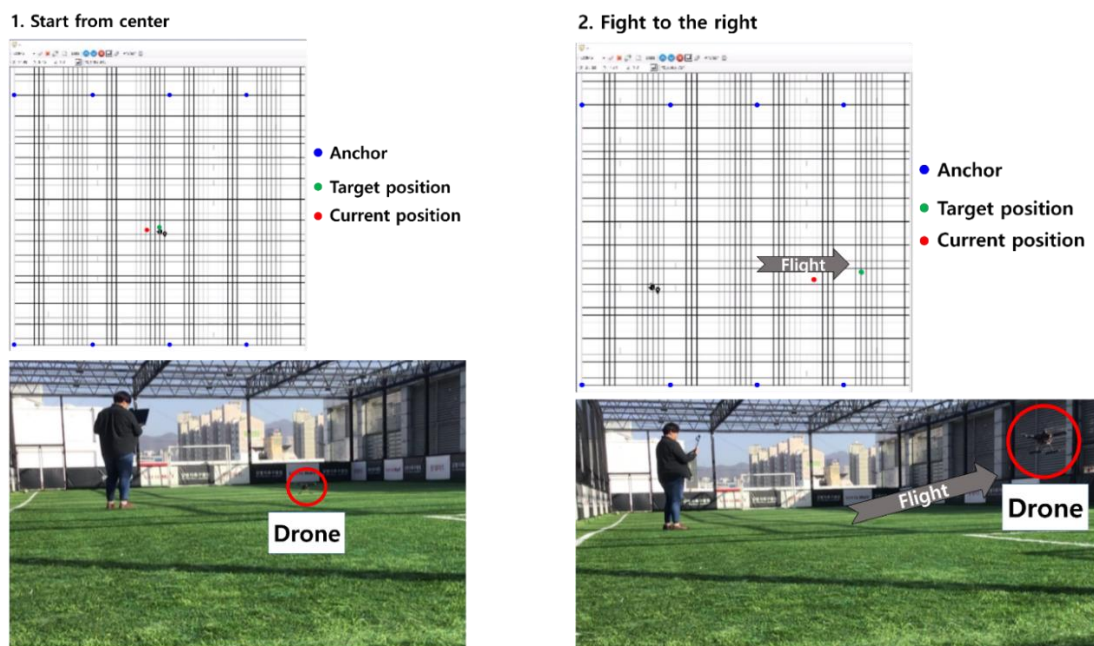


Figure 15. Waypoint flying demonstration for waypoint 1

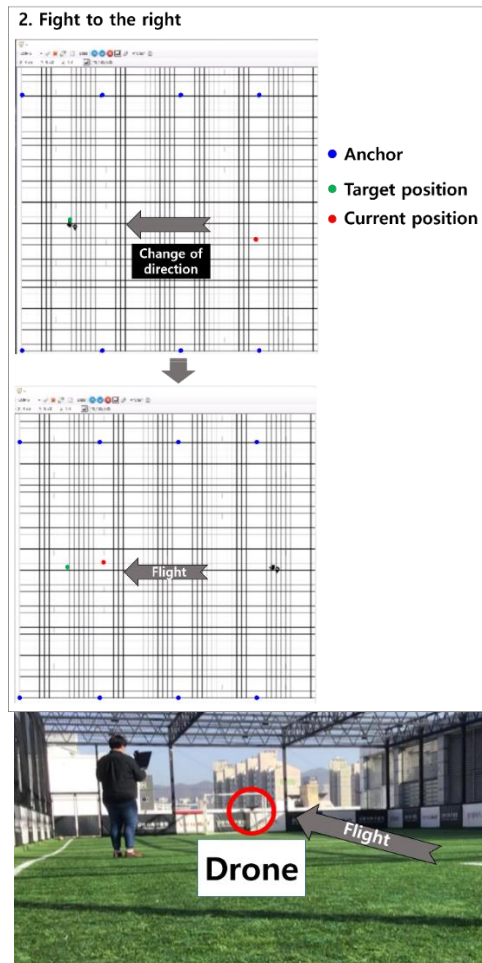


Figure 16. A 6-waypoint flying demonstration for waypoint 2

As a result of the experiments, the position error to the destination is: first, there is a 30 cm error caused by the DWM1000 module. In addition, by giving different flight commands for each section, the inertia force was weakened by the counter thrust to try to fly as stable as possible, but the inertia force could not be eliminated, so there is an average error of 20 cm due to inertia. Therefore, the error for the target position was shown to have a maximum radius of 50 cm, and we are sure that using a more accurate UWB module will have a smaller error radius. Despite this error, it was found that the waypoint flight was performed stably and relatively accurately. Table 1 shows the results of the way-point flying based on the proposed scheme, where the repetitive autonomous way-point flying has errors in the range of 30 cm at the x-axis and y-axis, respectively while maintaining an altitude of 1.5 m.

Table 1. Results of waypoint flying

Items	Unit	Value
Altitude	m	1.5
Drone flight speed	km/h	15
Position error	cm	<i>x-axis: 30; y-axis: 30</i>
Round trip distance	m	80
No. of round trips	times	13
Total flying distance	km	1.04

## 5. AUTONOMOUS INDOOR FLYING BY A SINGLE CAMERA ASSOCIATED WITH A DEEP LEARNING

In this study, a method using convolutional neural network (CNN) structure among AI has been proposed to carry out autonomous flying of drone with the camera image set. To determine if the autonomous

flight is possible from the input dataset of a single camera, the simulator was first employed to verify the performance, and the suitable architecture of deep learning was induced and later the effective results were produced through experiments.

### 5.1. Deep learning architecture design

Table 2 represents a CNN structure suitable for autonomously flying indoor drone using a single camera by applying the results of the simulator. The design is based on VGG16 [11] which is commonly used in image learning. Although large images are effective to better understand the characteristics of the images and to design the CNN structure in-depth, the input data size has been adjusted considering that the calculation should be within 4 ms in TX2. Table 2 describes the architecture of deep learning using a single camera, which shows a deep structure to ensure stable flying.

Table 2. CNN architecture of autonomously flying drone

Layers (Type)	Output Shapes	No. of Parameters
lambda_1(Lambda)	(None, 160,320,1)	0
conv2d_1	(None, 160,320,8)	80
conv2d_2	(None, 160,320,8)	584
max_pooling_2d_1	(None, 80,160,8)	0
conv2d_3	(None, 80,160,64)	4672
conv2d_4	(None, 80,160,64)	36928
max_pooling_2d_2	(None, 40,80,64)	0
conv2d_5	(None, 40,80,128)	73856
conv2d_6	(None, 40,80,128)	147584
max_pooling_2d_3	(None, 20,40,128)	0
conv2d_7	(None, 20,40,256)	295168
conv2d_8	(None, 20,40,256)	590080
conv2d_9	(None, 20,40,256)	590080
max_pooling_2d_4	(None, 10,20,256)	0
Dropout_2 (0.4)	(None, 10,20,256)	0
conv2d_10	(None, 10,20,512)	1180160
conv2d_11	(None, 10,20,512)	2359808
conv2d_12	(None, 10,20,512)	2359808
max_pooling_2d_5	(None, 5,10,512)	0
conv2d_13	(None, 5,10,512)	2359808
conv2d_14	(None, 5,10,512)	2359808
conv2d_15	(None, 5,10,512)	2359808
max_pooling_2d_6	(None, 2,5,512)	0
conv2d_16	(None, 2,5,1024)	4719616
max_pooling_2d_7	(None, 1,2,1024)	0
Dropout_2 (0.4)	(None, 1,2,1024)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 1024)	2098176
activation_1	(None, 1024)	0
dense_2 (Dense)	(None, 64)	65600
activation_2	(None, 64)	0
dense_3 (Dense)	(None,1)	65
Total parameters: 21,601,689		
Trainable parameters: 21,604,689		
Non-trainable parameters: 0		

The input data is followed by five convolution layers and is completed with a fully connected layer. The final layer was designed with a single node predicting the PWM duty value for the roll angle. Many channels were formed to understand the characteristics of the image. However, because of its deep design, numerous channels, and large size of input data, dropouts were set to 0.4 between layers in consideration of overfitting.

The mean square error is used as the loss function, Adam optimizer is used as the optimizer, and several rectified linear unit (ReLU) functions are used as the activation function [12]. Figure 17 shows the algorithm for controlling autonomously flying drones. The algorithm can be classified into three types. The first is to create a dataset to train. The second part is to find the optimal weights and biases by putting the dataset into a predesigned CNN structure. The third part is the real test using the CNN structure using the determined weights and biases.

First, the dataset creating part is: as mentioned earlier, the input dataset is composed of first-person view (FPV) images of the drone. The individual image size was set to (Cols, Rows)=(160, 320). As we aim to fly the drone as if it human pilot controls it, the label is set to be PWM duty value of the roll angle. The dataset was stacked in JSON format, which requires the following steps.

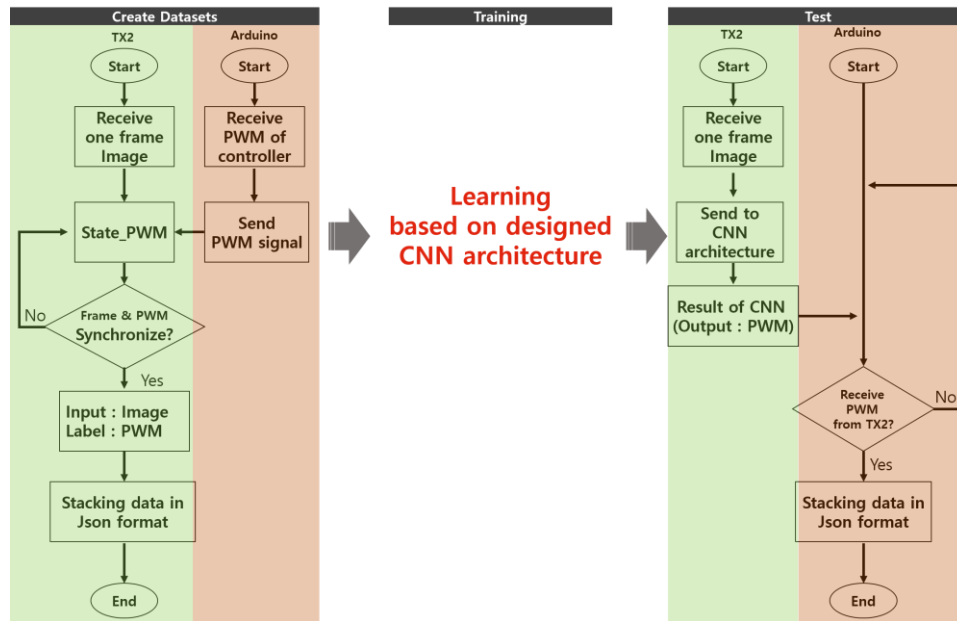


Figure 17. Algorithm of autonomous indoor flying with a single camera

First, take a frame of the FPV image from Jetson TX2 and receive the PWM value from the Arduino microcontroller. The two datasets received are stacked in JSON format by setting images as input data and PWM value as a label. Then, when all the datasets are gathered, we can use the JSON format dataset for the next training. The second part is the training part. In this part, the CNN structure is trained through the dataset created earlier, and then the weights and biases can be found. These weights and biases are finally used in the test process. The third part is the test part that finds the real-time PWM prediction value using the weights and bias found in the training through the CNN structure. Similarly, to create a dataset, a real time frame of FPV is taken and passed through the CNN structure to produce predicted PWM value, and then it is sent to Arduino. The PWM prediction value is sent to the FC to control the drone. It was proved that these algorithms enabled autonomous flight of hexacopter. Afterward, the consequences of autonomous flight will be described in the following section.

## 5.2. Experimental results of autonomous flying

The results of the autonomous flight of drone using the proposed CNN structure, which uses only a single camera are: the drone test was conducted in a 30 m long and 2.1 m wide corridor. The drone was controlled to fly while maintaining an altitude of 1 m. Figure 18 shows the result of the actual test of the drone and the images show a drone flying autonomously in the middle of the hall in a narrow corridor. However, due to the inherent drift of drone, it does not move forward, but slightly flies from side to side. At this time, FPV of the real-time drone comes in as an input dataset, so the corresponding PWM value of roll direction through the proposed CNN structure is predicted.

For every FPV image, the corresponding PWM duty value for the roll direction is shown in Figure 19. The PWM value is delivered to FC to fly autonomously. If the drone is located in the center of the hallway, CNN will output a PWM value of 500. If the drone is located on the left side of the hallway, the drone will predict the PWM value higher than 500 and fly to the right. In the same way, if the drone is on the right side of the corridor, it will predict the PWM value to be less than 500 and moves the drone to the left to stay centered. In the experiments, the PWM duty value is set to be between 0 and 1,000, but the threshold is set so that the PWM value is formed in the range of 100 to 800 to prevent the drone from tilting too much. However, because of the drone inertia, it was found that the drone moves from side to side while it keeps flying autonomously in a narrow hallway.

Table 3 shows the performance of the autonomous flying of drone using the proposed CNN structure that adopts a single camera, and it was found that the autonomous performance was satisfactory. As we can see from the result, the drone swings from side to side like a pendulum, flying forward while keeping the center of the hallway. Besides, we can see that the thresholds are applied at both ends so that the PWM values between 100 and 800 are transferred to the FC.

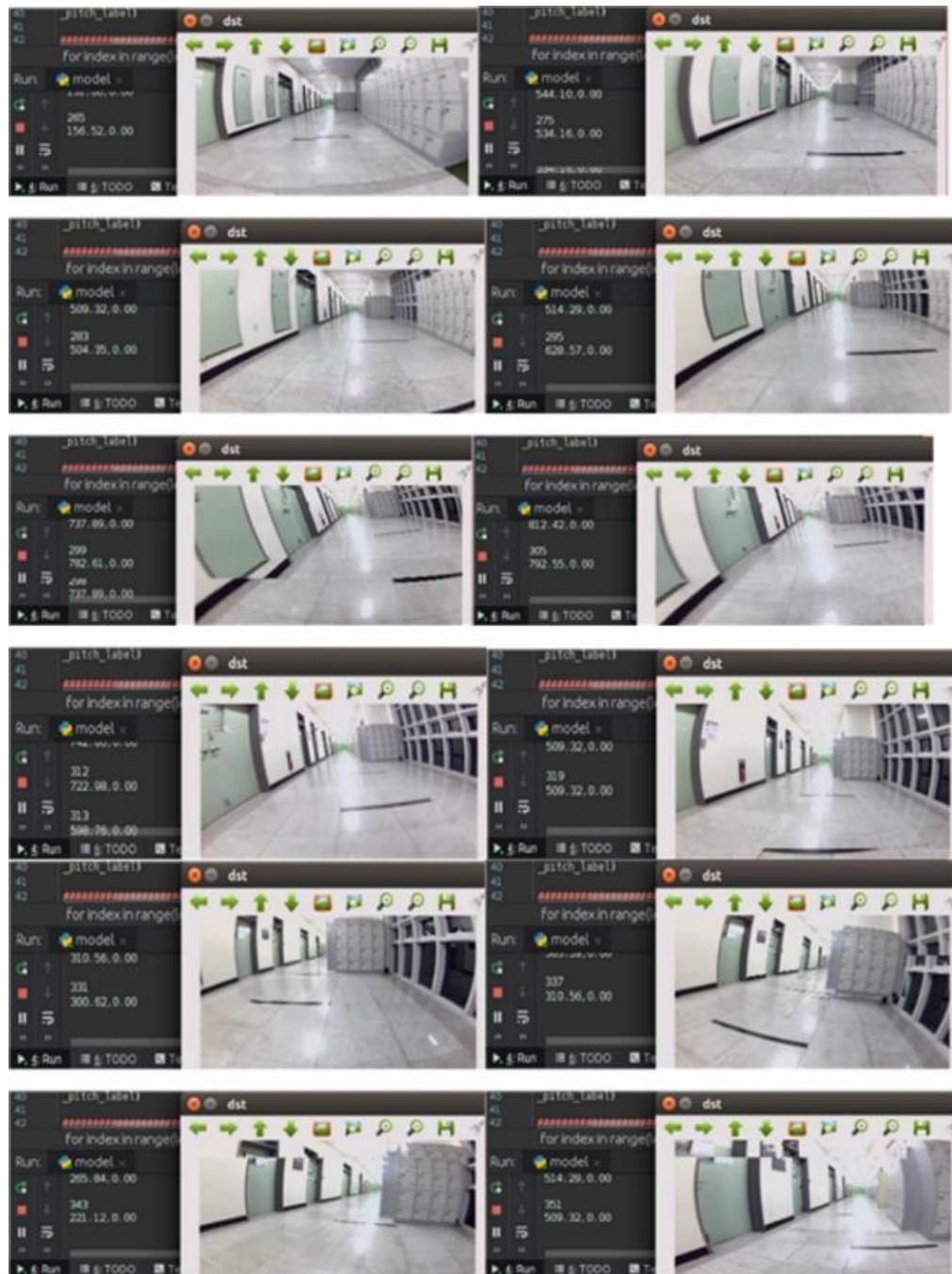


Figure 18. Image frames of autonomously flying drone at a corridor

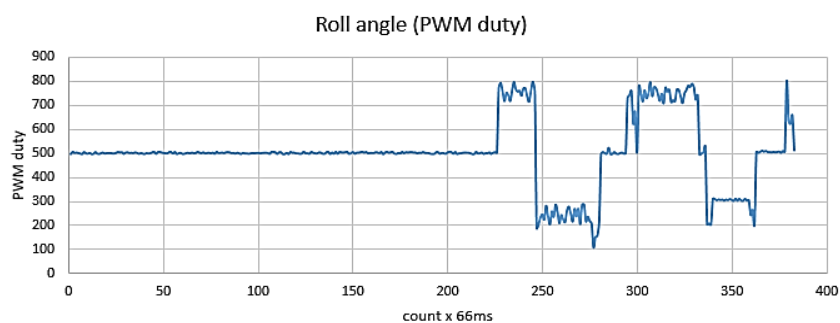


Figure 19. PWM value history of roll direction

Table 3. Experimental results from the proposed CNN structure

Items	Unit	value
Flight velocity	m/sec	1.4
Average deviation from the center	cm	12
Autonomous flying success rate to the destination	%	90

## 6. CONCLUSION

The work aimed to develop a drone that can fly safely in a specific environment where GPS cannot be operated due to lack of GPS operations such as tunnels and underground parking lots. In order to estimate the 3D position of the drone flying indoors and to control it, a DSP based control board and UWB modules were fabricated and control algorithms for a stable flight of the drone were developed. The 3D position estimation algorithm based on the proposed 2D projection method was implemented to obtain the 3D position of the drone, and the hovering motion control and way-point position control of the drone were performed using the estimated position information.

The attitude controller of the drone was designed employing a dual PID controller. The 9-axis IMU sensor used for attitude control is attached to the drone. The gains of all PID terms are determined through multiple experiments. DWM1000 modules were adopted to get the distance between anchor modules and a tag module which is mounted on the drone.

Since the drone is flying in three dimensions, three-dimensional position estimation based on the conventional gradient descent algorithm was originally tried. However, the inaccurate estimation of three-dimensional position frequently occurs due to sensor noise or problem of falling into a local minimum, which raises the issue that the existing three-dimensional position estimation algorithm needs to be modified. Here, an altitude measurement sensor was mounted to the drone to determine the altitude of the z-axis in a priori. By projecting the measured distance values between a tag and anchors onto a two-dimensional plane, which counts for x and y axes, and varies according to the drone altitude, we could reduce both the time required for solving the solution of estimating the location of the drone and the 3D position estimation error of the drone within 0.12 m.

Next, after placing four anchor modules to the wall of the indoor corridor and a tag module on the drone hovering over current location and waypoint flying to the target location were implemented. Two operations are successfully done, and it is found that 2D projection method along with measured altitude information is efficient to apply to indoor flying of drone under the denied GPS environment.

Applying the proposed 3D position estimation scheme and the control method to the designated position to the tunnel will help various rescue activities in tunnel accidents, and it will also be applicable to autonomous flight of an indoor flying object. The proposed structure of CNN was applied to the drone experimentally, and it was verified that the autonomous flight, which relies on only a single camera could be successfully performed even in the narrow corridor of the room. At this state, we state that if computer hardware grows rapidly, algorithms designed by deeper structure and big size input datasets using more cameras can accomplish complete autonomous flight.

## ACKNOWLEDGMENT

This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0008473, HRD Program for Industrial Innovation).





## REFERENCES

- [1] J. C. Lee, "Drones that expand the range of applications," *Magazine of the SAREK*, vol. 44, no. 11, pp. 90–91, 2015.
- [2] H. D. Yang, "Current status and technical trends of the private drone industry," *The Journal of multimedia information system*, vol. 20, no. 12, pp. 1–5, 2016.
- [3] H.-S. Moon, C. Kim, and W. Lee, "Drone-based location detection technique for buried persons in collapsed area," *The Magazine of the Korean Society of Civil Engineers*, vol. 64, no. 4, pp. 32–35, 2016.
- [4] J. H. Jin and G. B. Lee, "Understanding and trend of UAV/drone," *The Journal of The Korean Institute of Communication Sciences*, vol. 33, no. 2, pp. 80–85, 2016.
- [5] K. Miyazawa, "Fire robots developed by the Tokyo Fire Department," *Advanced Robotics*, vol. 16, no. 6, pp. 553–556, Jan. 2002, doi: 10.1163/156855302320535953.
- [6] K. Guo, Z. Qiu, W. Meng, L. Xie, and R. Teo, "Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in GPS denied environments," *International Journal of Micro Air Vehicles*, vol. 9, no. 3, pp. 169–186, Sep. 2017, doi: 10.1177/1756829317695564.
- [7] S. Zahran, A. Moussa, and N. El-Sheimy, "Enhanced drone navigation in GNSS denied environment using VDM and hall effect sensor," *ISPRS International Journal of Geo-Information*, vol. 8, no. 4, Apr. 2019, doi: 10.3390/ijgi8040169.
- [8] G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein, "GPS-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft," *Journal of Field Robotics*, vol. 30, no. 3, pp. 415–438, May 2013, doi:





- 10.1002/rob.21454.
- [9] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield, "Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 4241–4247, doi: 10.1109/IROS.2017.8206285.
  - [10] E. Unlu, E. Zenou, N. Riviere, and P.-E. Dupouy, "Deep learning-based strategies for the detection and tracking of drones using several cameras," *IPSJ Transactions on Computer Vision and Applications*, vol. 11, no. 1, Dec. 2019, doi: 10.1186/s41074-019-0059-x.
  - [11] S. Ross *et al.*, "Learning monocular reactive UAV control in cluttered natural environments," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1765–1772, doi: 10.1109/ICRA.2013.6630809.
  - [12] Y. Chen, P. Aggarwal, J. Choi, and C.-C. J. Kuo, "A deep learning approach to drone monitoring," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec. 2017, pp. 686–691, doi: 10.1109/APSIPA.2017.8282120.
  - [13] M. Segura, F. Auat Cheein, J. Toibero, V. Mut, and R. Carelli, "Ultra wide-band localization and SLAM: a comparative study for mobile robot navigation," *Sensors*, vol. 11, no. 2, pp. 2035–2055, Feb. 2011, doi: 10.3390/s110202035.
  - [14] E. Fresk, K. Ödmark, and G. Nikolakopoulos, "Ultra wideband enabled inertial odometry for generic localization," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11465–11472, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.1820.
  - [15] C. E. Lee and T. G. Seong, "Introduction of UWB positioning technology and technical trends," *The Journal of The Korean Institute of Communication Sciences*, vol. 34, no. 4, pp. 33–38, 2017.
  - [16] B. K. Cheon and K. J. Yoon, "Attitude controller design of small tilt multicopter drone," *Korean Society for Aeronautical and Space Sciences Conference*, pp. 128–129, 2017.
  - [17] E. Sin, "Controller design and experiment for attitude recovery of a quadrotor," *Academic Journal of KSAS*, pp. 386–389, 2015.
  - [18] H. S. Yang and D. J. Lee, "Basics of drone flight control, and status estimation," *The Journal of The Korean Institute of Communication Sciences*, vol. 33, no. 2, pp. 86–92, 2016.
  - [19] G. H. Cha, I. Sim, S. G. Hong, J. H. Jung, and J. Y. Kim, "A study of method and algorithm for stable flight of drone," *Journal of Satellite, Information and Communications*, vol. 10, no. 3, pp. 32–37, 2015.
  - [20] "Flight controller selection." [https://docs.px4.io/main/en/getting\\_started/flight\\_controller\\_selection.html](https://docs.px4.io/main/en/getting_started/flight_controller_selection.html).
  - [21] Y. Zhou, "An efficient least-squares trilateration algorithm for mobile robot localization," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 3474–3479, doi: 10.1109/IROS.2009.5354370.

## BIOGRAPHIS OF AUTHORS







**Seung Je Son**     graduated from Department of Mechanical System Design Engineering at Seoul National University of Science and Technology in 2017 in Korea. Also, he received M.S. from Department of Mechanical Design and Robot Engineering from the same university in 2019. His major research interests are robot vision, deep learning, and system integration. After graduation he is working at Betteral Ltd, Korea as a researcher. He is doing research on embedded automation, artificial intelligence, and equipment control. He can be contacted at [ssj@betteral.io](mailto:ssj@betteral.io).



**Hun Se Kim**     graduated from Department of Mechanical System Design Engineering at Seoul National University of Science and Technology in 2016 in Korea. Also, he received M.S. from department of Mechanical Design and Robot Engineering from the same university. Currently, he is working at Truen Ltd. in Korea as a junior engineer. His major research interests are CCTV Firmware, Embedded Programming, and Intelligent System. He can be contacted at [k930410@naver.com](mailto:k930410@naver.com).



**Dong Hwan Kim**     received B.S and M.S from Department of Mechanical Design and Production Engineering at Seoul National University in 1986 and 1988. He received a Ph.D. from Georgia Institute of Technology, USA. He worked at Korea Institute of Industrial Technology from 1997-1998. He joined Seoul National University of Science and Technology in 1998 as a professor at Department of Mechanical System Design Engineering. His major research interests are robot control, deep learning, and mechatronics. He is doing numerous projects on robot mechanism and control, artificial intelligence applications to robot, and smart mechatronics system. He can be contacted at [dhkim@seoultech.ac.kr](mailto:dhkim@seoultech.ac.kr).