Master's Thesis
석사 학위논문

Development of a Quadrotor-type UAV Platform for Autonomous Collision Avoidance under Teleoperation

SangYong Park(박 상 용 朴 桑 用)

Department of
Information and Communication Engineering

DGIST

2020

Master's Thesis
석사 학위논문

Development of a Quadrotor-type UAV Platform for Autonomous Collision Avoidance under Teleoperation

SangYong Park(박 상 용 朴 桑 用)

Department of
Information and Communication Engineering

DGIST

2020

Development of a Quadrotor-type UAV Platform for Autonomous Collision Avoidance under Teleoperation

Advisor: Professor Kyoung-Dae Kim
Co-advisor: Professor Jaeha Kung

by

SangYong Park
Department of Energy Science & Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information and Communication Engineering. The study was conducted in accordance with Code of Research Ethics[1]

12. 26. 2019

Approved by

Professor Kyoung-Dae Kim (signature)
(Advisor)
Professor Jaeha Kung (signature)
(Co-Advisor)

---

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Development of a Quadrotor-type UAV Platform for Autonomous Collision Avoidance under Teleoperation

SangYong Park

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

11. 26. 2019

Head of Committee	Prof. Kyoung-Dae Kim(signature)
Committee Member	Prof. Jaeha Kung (signature)
Committee Member	Prof. Sunghyun Cho (signature)

MS/IC
201822007

박상용. SangYong Park. Development of a Quadrotor-type UAV Platform for Autonomous Collision Avoidance under Teleoperation. Department of Information and Communication Engineering. 2020. 39p. Advisors Prof. Kyoung-Dae Kim, Co-Advisors Prof. Jaeha Kung

ABSTRACT

In recent years, unmanned aerial vehicle (UAV) systems have been used successfully in many tasks such as search and rescue, remote sensing, mapping, exploration, surveillance, and many other civil and military applications. In general, an unmanned aerial vehicle is a powered aircraft that can be operated remotely or automatically without human boarding. Therefore, it has an advantage over general aircraft in terms of size and weight, so it can be usefully used for various tasks as described above.

Accordingly, the development of unmanned aerial vehicle technology and the growing demand for unmanned aerial vehicles are expected, and various types of high performance unmanned aerial vehicles have been developed and launched. As the use of unmanned aerial vehicles has soared, there has been a growing interest in unmanned aerial vehicle collision avoidance technology as concerns about collisions with buildings, aerial installations, and even collisions with airplane. However, as one may know, it is actually not a trivial task to remote control a UAV safely, especially in a cluttered environment. Hence, autonomous collision avoidance is considered as one of the essential capabilities that UAVs must provide.

And also, the autonomy level of robotic system is still restricted by the deficiency of a robust and reliable perception, and of a higher cognitive ability that allows sophisticated decision making in real world environment. This is especially true for robots that have high degrees of freedom such as UAV. Thus, in many cases, human supervisory is still required to perform high level decision making while UAVs execute their local autonomy such as obstacle avoidance. Therefore, to ensure that the UAV safely follows the human operator's command, the high level of decision-making that can be performed by the human operator and the proper integration of local autonomy that the UAV can perform on its own are essential. The representative local autonomy that UAV can perform is the autonomous collision avoidance of the UAV itself. Therefore, there is no doubt that autonomous collision avoidance is indeed one of the essential capabilities that a UAV should have for the sake of UAV operational safety. Therefore, in this thesis, present a highly reliable autonomous collision avoidance algorithm, Vehicle-Centered Potential Function (VPF), and verify the performance through extensive simulation using the robot simulation software V-REP. After successfully verifying the VPF-based autonomous collision avoidance algorithm through simulation, need to verify the performance on a real UAV platform. Therefore, as part of an extensive research project on autonomous collision avoidance of remotely operated UAVs, this particular study focuses on the development of a real UAV platform and uses it to evaluate collision avoidance performance through experiments. More specifically, designed both UAV's hardware and software systems, including ground control systems. In addition, sensors are mounted on UAV for object detection and identification and a high confidence recognition system is established through sensor fusion. Through the above researches, built an autonomous collision avoidance system of UAV and verify the performance through experiments.

Keywords: UAV, Autonomous collision avoidance, VPF, Sensor fusion

List of Contents

List of equations

List of figures

I. Introduction

In recent years, multirotor-type unmanned aerial vehicles (UAVs) have been successfully used for many tasks such as search and rescue, reconnaissance, remote sensing, mapping, surveillance, target acquisition, border patrol, aerial imaging, industrial inspection, infrastructure monitoring, emergency medical aid, exploration, and many other civil and military applications thanks to their various advantages such as size, cost, weight, and, more importantly, the ability of vertical take-off [1]-[3]. Some UAVs can perform increasingly complex autonomous maneuvers, but most UAVs are not completely autonomous. Instead, they are mainly operated remotely by humans. However, as one may know, remote control of a UAV is not a trivial task, especially in a cluttered environment. Typically, it requires a high level of expertise as well as enough training to teleoperate a multirotor-type UAV safely. However, even for an expert, it is still a challenge to remote control a UAV safely since the operator is physically separated from the UAV by a certain distance which inevitably makes the operator to have poor situation awareness about the surroundings where a UAV is flying. A lot of technology and algorithm development is still needed to make UAV completely autonomous. For example, UAVs need to improve obstacle detection and subsequent avoidance. This is especially important as autonomous UAVs begin to operate in the civil aviation space used by other aircraft. It is useful to operate Operating unmanned flying vehicles, but it can be difficult when the unmanned flying vehicles such as UAV interacts with the environment. This interaction can be in the form of, for example, landing on a ground or landing pad, docking to a station, accessing terrain for inspection, or approaching another aircraft for refueling. These tasks can often be solved when the UAV is remotely controlled, especially when the pilot has a first-person view of the environment. Therefore, it is important to find an effective and flexible strategy for the UAV to perform these tasks autonomously.

As for the reasons mentioned above, there is no doubt that autonomous collision avoidance is indeed one of the essential capabilities that a UAV should have for the sake of UAV operational safety. In literature, various approaches have been proposed to develop techniques for autonomous collision avoidance of UAV [4]-[11]. However, to the best our knowledge, none of these works provides capabilities desirable for autonomous collision avoidance of a teleoperated UAV such that a teleoperated UAV can avoid collisions with multiple moving objects simultaneously, and a UAV can still avoid collision via an autonomous evasive motion even when there is no operator input or all the collective collision avoidance motion fails to avoid collision. Recently, a novel approach, called the Vehicle-centered Potential Function (VPF), that is capable of provisioning all above mentioned desirable capabilities is developed in our research group and its performance has been validated through extensive simulations using a robot simulation software V-REP.

After successful validation of the VPF based collision avoidance algorithm through simulations, now it is necessary to validate its performance in a real UAV platform. Therefore, as a part of a broader research project on autonomous collision avoidance of a teleoperated UAV, this particular research focuses on the development of a real UAV platform and evaluates the collision avoidance performance through experimentation using it. More specifically, we design both the hardware and software system of a UAV including the ground control system. In software system, we also need to design and implement all necessary functionalities for perception, navigation via remote control, and, of course, VPF-based collision avoidance. Since there was no need for perception in simulations, it is necessary for us to additionally design and implement software modules for static/dynamic object detection and tracking. Hence this is another focus in this research.

In summary, this research is about the development of a UAV platform for autonomous collision avoidance as well as the sensor fusion to improve the overall object detection quality. In the following sections, we briefly describe our design and the current state of overall development process. In last section of this proposal, we also summarize what are to be done more to complete the development and plans for experimental validation.

II. Related Work

The most important technologies when utilizing drones such as exploration, surveillance, search mentioned so far are those directly related to safety. In particular, the military mission of hundreds to thousands of drones requires avoidance technology that prevents them from colliding with each other and technology that can avoid even the smallest obstacles by itself. To this end, many technologies are being studied for autonomous flight as well as the autonomous collision avoidance system of UAVs. There are many studies required for autonomous flight of UAVs, but the most important and necessary studies can be divided into two main categories. First, the study of autonomous collision avoidance algorithms. Second, it consists of a study on the environmental perception of UAVs in flight using sensors.

As mentioned earlier, the collision avoidance algorithm for autonomous flight of unmanned aircraft has been studied and is still being studied by many scholars around the world. The paper of Novell 3D geometric algorithm on aircraft automatic collision avoidance [12] presents a new determination algorithm for non-cooperative aircraft airborne collision avoidance. The reasons why the algorithms presented in this paper are suitable for collision avoidance algorithms in unmanned aircraft do not require the resolution of programming problems. Therefore, it is suitable for many other real-time applications as well as unmanned aircraft. The algorithms proposed in this paper minimize aircraft deviation in nominal trajectory and perform optimal avoidance maneuvers in both horizontal and vertical planes. Numerical simulations have proven effective in appropriate crash scenarios, considering aircraft dynamics and in-flight sensor limits. and, in the paper [13], researched the automatic generation of collision avoidance algorithms given models of aircraft dynamics, sensor performance, and intruder behavior. This paper showed that the MDP/POMDP formulation is flexible enough to accommodate various sensor forms, intruder behavior, aircraft dynamics, and cost functions.

As well as collision avoidance algorithms for safe flight of UAV, perception systems based on sensors that can be mounted on UAV are also actively being studied. Intelligent robot development, including unmanned aerial vehicles, requires several types of sensor technology. A distance sensing system for robot collision avoidance, obstacle detection, etc. will be required, a positioning system for robot positioning, and a sensing system for robot visual information acquisition. In [14], LiDAR sensor was used to investigate dense forest structure. LiDAR sensors are sensors that have sensing methods that detect objects and map distances. Measure the characteristics of the reflected return signal after it is illuminated by an optical pulse. The width of the optical pulse can vary from a few nanoseconds to a few microseconds. The LiDAR sensor is used in many robot platforms because of its short frequency, it can easily detect even small objects and create accurate 3D monochrome images [15]-[18]. However, use is limited in night and cloudy weather, and the operational altitude is limited to 500-2000 meters. And there is a cost burden because it is relatively more expensive than other sensors. And most of all, there is a lack of visibility. On the other hand, visual sensors play a pivotal role in robot recognition systems that provide flexible information to cope with changing environments. Therefore, visual sensors can be suitable for platforms where visibility is critical, such as UAV systems. Visual sensors acquire images in real time, mainly using cameras, and process vast amounts of acquired image data in real time to analyze information such as the size, location, color, etc. of objects from images and recognize objects. Based on a general mono vision, there is also a paper that studies various platforms' cognitive systems and mobile robots' navigation by applying technologies such as SLAM [19]-[21]. More recently, studies have been underway to extract real-time three-dimensional image information of objects using more than one of these visual sensors. The production technology of real-time distance images can be applied as component technology in many areas such as movies, animations, broadcasting, and virtual reality, as well as applications in intelligent robot fields such as autonomous driving of robots and obstacle recognition. The sensors that enable the above role are stereo cameras and are also used on many robot platforms [22]-[25]. Stereo cameras allow users to obtain depth. information that cannot be obtained by using regular mono vision, so they can implement technologies such as local mapping as well as real-time spatial recognition. However, not only stereo cameras, but many other visual sensors cannot get 'velocity' information about objects. In the case of autonomous robot systems that require accurate location and speed information for surrounding objects, such as UAV systems, velocity information is critical. The radar sensor uses FMCW (Frequency Modulated Continuous Wave) radar to reliably detect moving or stationary targets, including cars, trains, trucks and cargo, even in extreme weather conditions. and, radar's Doppler Effect provides speed information for the relative object. The Doppler Effect of radar sensors, which can measure microwave signals against targets and then analyze how the frequency of returned signals has changed with the movement of objects and extract speed information of objects, has been studied a lot on a number of autonomous robot platforms [26]-[28]. However, the disadvantages of radar sensors are not to recognize the shape of objects, and it is difficult to increase the measured distance and angle at the same time. As mentioned above, each sensor has its own limits, which can be a major problem in designing the robot’s autonomous system. As discussed earlier, it is necessary to combine the advantages of each sensor because each sensor has its role and advantages and disadvantages. Since safety is the most important factor in a robot’s autonomous driving and UAV’s autonomous flight systems, clear recognition is needed even in conditions that are difficult to recognize, such as night, snow and strong backlight. For example, if fog is severe, information obtained from the camera is inaccurate and can be safely determined through information such as LiDAR or radar. Therefore, in order to supplement the unique problems of one sensor and increase the accuracy of the recognition system, sensor fusion technology that combines multiple sensors into one sensor is being studied [29]-[32]. As such, there are several requirements (autonomous collision avoidance, sensor-based object detection and tracking technology) required to build the UAV’s autonomous system [33]. as an extension study of this system, in this paper, a medium-sized UAV platform is built to establish an autonomous flight system of unmanned aircraft. In addition, sensors are mounted on unmanned aircraft to study sensor-based cognitive systems and high confidence autonomous collision avoidance algorithms.

III. Configuration of a UAV Platform

In order to build an autonomous collision avoidance system for a UAV, it is first necessary to construct an appropriate medium-sized UAV platform. Commercially available UAVs vary in size and spec depending on their purpose. But most of them are small UAVs for leisure and Video shooting purposes and they are not suitable for developers.

To install a computing board and sensors on UAV, need to provide the appropriate payload, and also provides a Software Development Kit (SDK) to set up the UAV system for developer purposes. Therefore, considering a variety of factors, including payload and experimental environment, we chose the MATRICE 100(M100), designed by DJI for developers (Figure 1). The most notable thing about the M100 is that it was purpose-built to be modified by developers for specific uses. Using its hardware expansion bays, the M100 can be configured to carry any set of sensors or devices that developer wants to put into the sky. DJI SDK allows developers to build custom mobile apps and advanced flight controls for any requirement.

There is a plate on the top of the UAV body and provides a suitable payload. However, since only the basic UAV body is provided, it is necessary to set up a hardware and also software system for the purpose of UAV usage. In this section, we describe how our UAV platform is configured in both hardware and software for autonomous sense-and-avoid capability.

	Specification
Production	DJI
Weight	2.3kg
Frequency	2.4GHz
Battery capacity	22.2V , 4500mAh
Flight distance	2km
Payload	3.4 kg

Figure 1. MATRICE 100, a DJI’s UAV platform for developer and hardware specification

3.1 Hardware system

In order to build an autonomous collision avoidance system for a UAV, a medium-sized UAV platform that can be operated outdoors is needed first. Therefore, as mentioned above, we selected DJI's MATRICE 100, a medium-sized UAV designed for developers. However, even if the UAV is designed for developers, it is essential to configure additional hardware for the developer's purpose. Since we had to mount various parts on the UAV, we needed extra space besides the top space provided by the MATRICE 100. Therefore, additional space was secured by manufacturing a separate plate and mounting a separate plate on the MATRICE 100 top plate. Therefore, separate plates consisting of three layers are equipped with components such as sensors and computing board. In more detail, the hardware system is largely composed of three parts, the aircraft part, the sensor part, and the ground controller part. The sensor part is composed of stereo vision and radar sensors, which are sensors used for object detection and tracking, It is located on the 3rd floor of the plate as shown in the Figure 2. The second floor consists of an onboard computer for sensor data and collision avoidance algorithm calculations, and the first floor consists of a battery to power the sensors and the onboard computer. And the aircraft part is composed of an onboard computer and a flight controller that can calculate the collision avoidance algorithm based on sensor data. Finally, the ground controller part consists of a remote controller and tablet PC, allowing an operator to check flight status and aircraft status in real time. Since there is not enough space to mount components such as onboard computers, sensors, etc. on the top plate provided by the MATRICE 100, we designed and installed an additional mounting module that consists of three layers of plate. (Figure 2) The first layer contains a battery that powers the onboard computer, the second layer contains an onboard computer, and the third layer contains stereo vision and radar. As for the onboard computer, we chose the NVIDIA JETSON TX2 considering the weight, capacity, and performance for necessary computations.

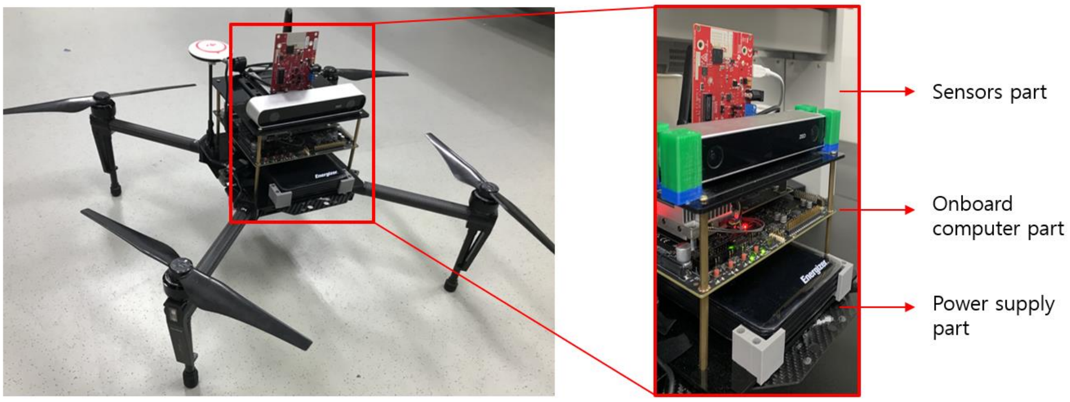


Figure 2. Three layers of plate manufactured separately and mounted on top of M100

3.2 Software system

The hardware and software systems consist of two parts, the aircraft part and the ground controller part. (Figure 3) First, the aircraft part consists of an onboard computer for calculating sensor data or collision avoidance algorithms. For the onboard computing board, we chose NVIDIA's Jetson TX2 considering the size and payload of the M100 platform. The Jetson TX2 is an embedded AI computing device that offers exceptional speed and power efficiency. It also includes a GPU with a wide range of standard hardware interfaces that are perfectly suited for a wide variety of products and form factor, enabling deep learning based algorithm calculations. Onboard computer systems receive sensor data from stereo vision and radar sensors and perform tasks such as object detection and tracking and 3D local mapping. In addition, the flight status is updated in real time from the flight controller of the UAV. The onboard computer integrates and computes the above information and finally sends the collision avoidance control input to the UAV's flight controller. In addition, the ground controller part allows an operator to send remote control commands to the UAV and also view the UAV's current flight data (GPS data) and aircraft status in real time through the tablet PC attached to the remote controller. One notable customization in our UAV software system is to make the user commands transmitted from the remote controller delivered to the onboard computer directly bypassing the UAV’s flight controller as the default configuration of the MATRICE 100’s flight controller is to response immediately upon receiving commands from the remote controller. With this customization, it is now possible to make the onboard controller become the main controller for the UAV movement.

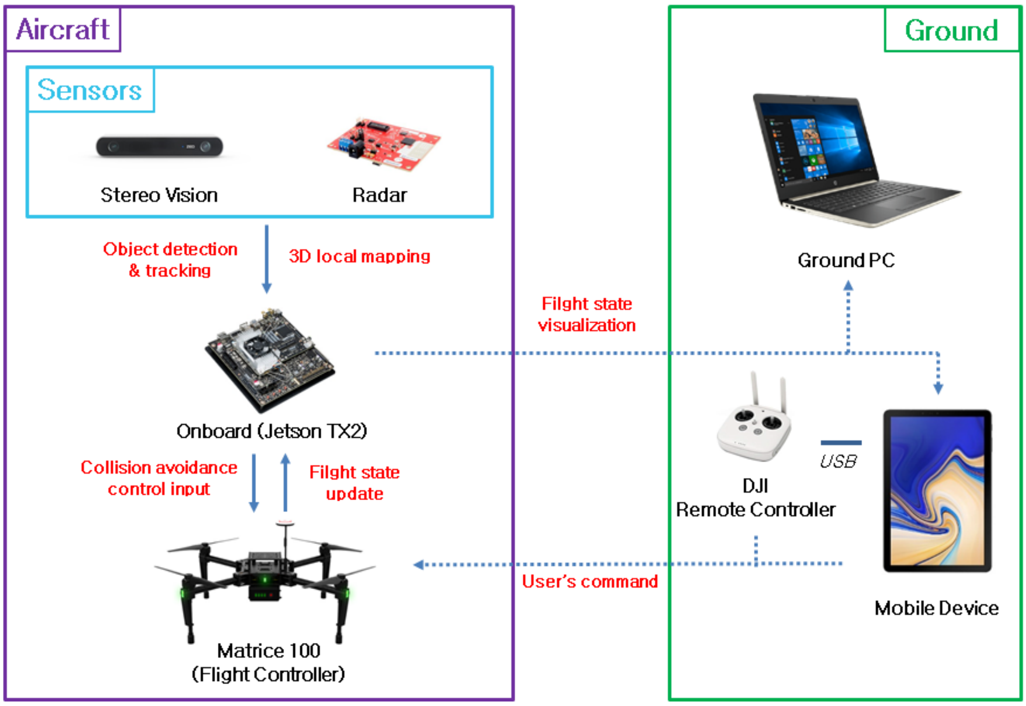
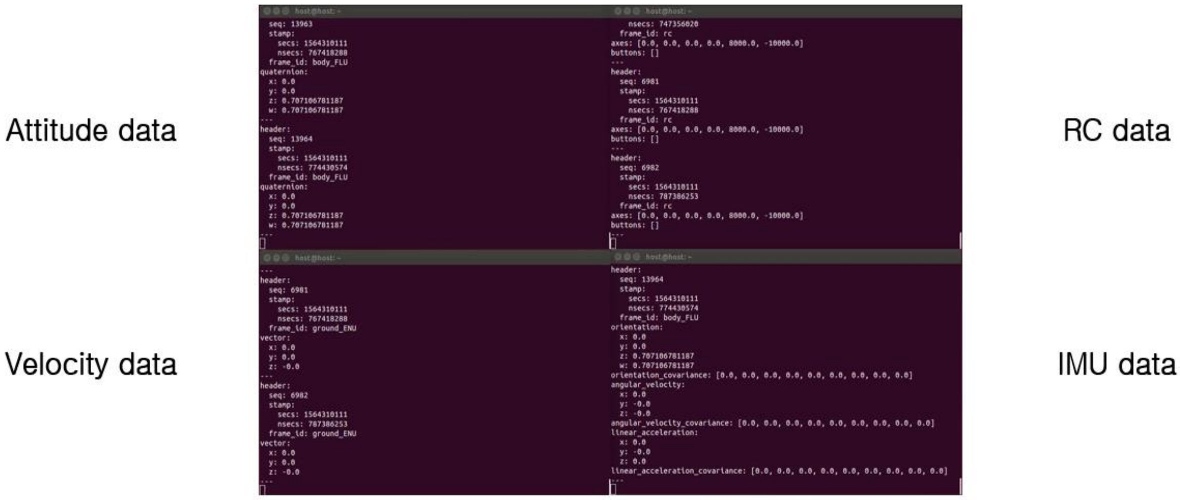


Figure 3. Hardware and Software systems architecture

In more detail, in a typical UAV system, all the operator's commands are set to be sent directly to the UAV's FC. Also, for the most part, this FC is not an open-source program, so the operator cannot set it up arbitrarily. Thus, above mentioned vertical structure does not allow developers to add functions or receive mounted sensor data. Therefore, it is essential to change authority from FC to the onboard computer. With the above operation, the operator's command is not sent directly to the FC, but to the onboard computer, which is then sent to the FC. Therefore, the final output with the system that the developer wants is transmitted to the FC from the onboard computer. For the above work, DJI provides SDK. SDK is a tool that simplifies the application development process by managing lower-level functions such as flight stabilization, battery management, signal transmission and communication. It is largely divided into OSDK and MSDK. First, the OSDK is the Onboard SDK, a software development kit that developers need to build an on-board system. It is possible to control UAV from onboard computers that are mounted through OSDK. Secondly, MSDK is a Mobile SDK software development kit designed to enable developers to access the functions of the DJI UAV product. This allows users to check the real-time location of the UAV using Google map and to give additional commands to the UAV (waypoint, hotpoint mission etc.). in other words, tablet PC can access controllers and on-board systems using the MSDK and receive and visualize the GPS status of UAVs in flight in addition to direct control through the controller.

Therefore, in this thesis, the software system is constructed as above. Add to, an experiment was conducted outdoors to check whether the computing board received a driving command using a remote controller on the ground or a Waypoint command using a mobile device and applied a control input to the UAV'S flight controller. The test confirmed that the UAV was operating normally for both situations. Figure 4 shows that the aircraft status of the UAV in flight can be checked on the onboard computer. Figure 5-(a) is the situation where the operator on the ground enters the flight command through the remote controller, and the computing board of the UAV in flight receives and controls the command. Figure 5-(b) shows a situation where a UAV is flying by receiving a current location of a UAV using a mobile device and transmitting a waypoint mission to the UAV.



```
  seq: 13963
  stamp:
    secs: 1564310111
    nsecs: 767418288
  frame_id: body_FLU
quaternion:
  x: 0.0
  y: 0.0
  z: 0.707106781187
  w: 0.707106781187
---
header:
  seq: 13964
  stamp:
    secs: 1564310111
    nsecs: 774430574
  frame_id: body_FLU
quaternion:
  x: 0.0
  y: 0.0
  z: 0.707106781187
  w: 0.707106781187
---
```

```
    nsecs: 747356020
  frame_id: rc
axes: [0.0, 0.0, 0.0, 0.0, 8000.0, -10000.0]
buttons: []
---
header:
  seq: 6981
  stamp:
    secs: 1564310111
    nsecs: 767418288
  frame_id: rc
axes: [0.0, 0.0, 0.0, 0.0, 8000.0, -10000.0]
buttons: []
---
header:
  seq: 6982
  stamp:
    secs: 1564310111
    nsecs: 787386253
  frame_id: rc
axes: [0.0, 0.0, 0.0, 0.0, 8000.0, -10000.0]
buttons: []
---
```

```
---
header:
  seq: 6981
  stamp:
    secs: 1564310111
    nsecs: 767418288
  frame_id: ground_ENU
vector:
  x: 0.0
  y: 0.0
  z: -0.0
---
header:
  seq: 6982
  stamp:
    secs: 1564310111
    nsecs: 787386253
  frame_id: ground_ENU
vector:
  x: 0.0
  y: 0.0
  z: -0.0
---
```

```
header:
  seq: 13964
  stamp:
    secs: 1564310111
    nsecs: 774430574
  frame_id: body_FLU
orientation:
  x: 0.0
  y: 0.0
  z: 0.707106781187
  w: 0.707106781187
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: 0.0
  y: -0.0
  z: -0.0
angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
  x: 0.0
  y: -0.0
  z: 0.0
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
---
```

Figure 4. Flight status check of the UAV on onboard computer

(a) **(b)**

Figure 5. (a) Flight situation 1. Transferring flight command to onboard system via RC (b)Flight situation 2. Receiving a waypoint flight command from the mobile device

3.3 Custom Software System for Collision Avoidance

There are many computational tasks that should be run on the UAV’s onboard computer such as sensor data acquisition and fusion, object detection and tracking algorithm, collision avoidance algorithm, and data transmission between the onboard computer and the flight controller as well as the ground control system. In our implementation, software system for all of these functionalities is developed based on the Robot Operating System (ROS) environment.

Roughly speaking, there are five modules in our software system configuration, that are sensing, perception, collision avoidance, UAV, and ground control. First, the sensing module consists of two ROS nodes, ‘zed_node’ and ‘radar_node’, that interface with attached sensors. The perception module is composed of ROS nodes for object detection and tracking and also mapping. Through the perception module, the information for collision avoidance is extracted such as the pose and velocity of detected objects. The collision avoidance module has only one ROS node, named ‘collision_avoidance_node’, which takes inputs from an operator via the ground controller, the perception module, and also the flight state information from the UAV’s flight controller. In our implementation, the software module that we have implemented for ground control is based on the mobile SDK provided by DJI for MATRIE 100. Finally, the UAV module is responsible for exchanging information between the collision avoidance module and the UAV's flight controller. Figure 6 shows the overall configuration of our custom software system for UAV collision avoidance.

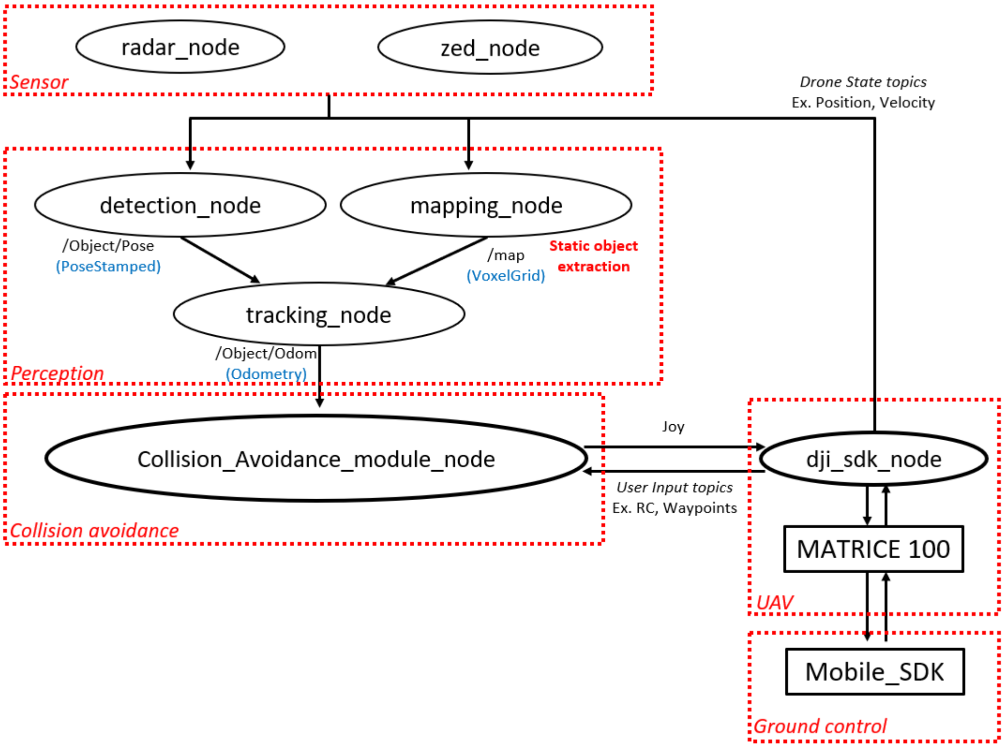


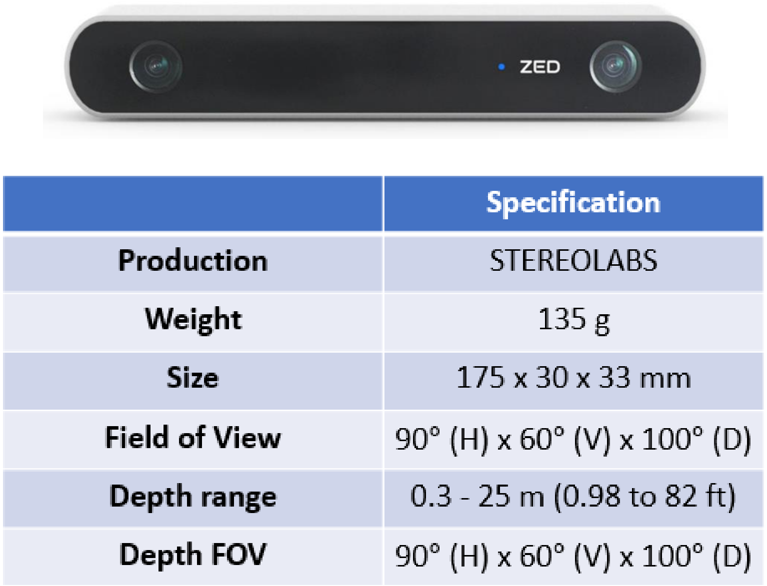
Figure 6. Custom software system for collision avoidance

IV. Object Detection and Tracking

In general, as each sensor has its own characteristics and limitations in terms of accuracy, sensing range, sensitivity to weather conditions, etc., it is very important to select an appropriate suite of sensors for the purpose of application in order to maximize the overall perception quality by compensating the disadvantages of each sensor [34]-[37]. In this section, we first describe briefly on our selection of sensors for the purpose of collision avoidance and presents some details on the perception modules that we are currently developing based on the data from the selected sensor suite.

4.1 Sensor Fusion

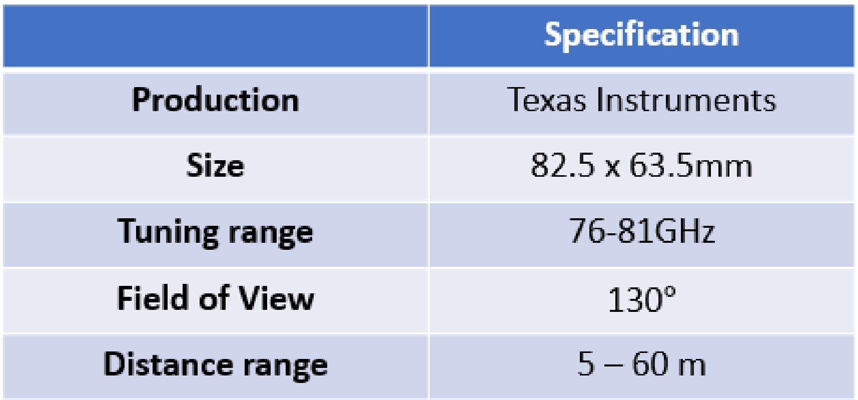
For medium-sized UAV platform such as MATRICE 100, there are only a few types of sensor to choose from due to various issues such as payload, size, power, etc. In this research, we chose stereo vision and radar sensor as suitable sensors for our UAV platform. Compared to the monocular vision system, the stereo vision system can determine the distance to an object by obtaining a stereoscopic image using two lenses. Knowing depth information is a big advantage for the purpose of collision avoidance. However, since the sensing range of stereo vision is generally short, it is difficult to detect objects approaching from medium to long distance. Furthermore, the speed of moving objects has to be estimated if only the stereo vision is used for detection. Therefore, we decided to fuse the stereo vision system with a radar sensor in order to compensate for the shortcomings of the stereo vision and improve the overall detection performance on moving objects. As shown in Figure 7, 8, our specific choice of sensors installed in the UAV platform are STEREOLABS's ZED stereo vision and TI's IWR1642BOOST radar sensor. The ZED stereo camera has three main advantages. First, long-range 3D sensing. sensing distance of ZED camera is possible up to 20 meters, which is enough distance to detect objects approaching the UAV. And, Dual 4MP camera. The ZED stereo camera provides 2K 3D video capture with low-light sensitivity to operate in the most demanding environments. It can also clearly identify objects approaching the UAV. Finally, High frame-rate. it can possible capture 1080p HD video at 30FPS or WVGA at 100FPS and get a clear image. The resolution of images is a significant factor because they are directly related to detection accuracy. Therefore, a zed stereo camera with a high frame rate is suitable as a stereo vision for detecting objects. The Figure 7 below shows a simple specification of the ZED camera.



	Specification
Production	STEREOLABS
Weight	135 g
Size	175 x 30 x 33 mm
Field of View	90° (H) x 60° (V) x 100° (D)
Depth range	0.3 - 25 m (0.98 to 82 ft)
Depth FOV	90° (H) x 60° (V) x 100° (D)

Figure 7. Hardware specification of ZED

And the model name of the radar sensor chosen as another sensor for sensor fusion is IWR1642BOOST from Texas Instruments™. The IWR1642Boost radar pack includes everything necessary to start developing software for on-chip C67x DSP cores and low-power arm® R4F controllers, including on-board emulation for programming and debugging and on-board buttons and LEDs for rapid integration of simple user interfaces. It has a wide range of sensing and can obtain accurate object location and Doppler effect, so it is suitable as a sensor for the UAV system. The Figure 8 below shows a simple specification of the IWR1642BOOST radar sensor



	Specification
Production	Texas Instruments
Size	82.5 x 63.5mm
Tuning range	76-81GHz
Field of View	130°
Distance range	5 – 60 m

Figure 8. Hardware specification of IWR1642BOOST

In order to fuse data from a radar sensor and a stereo vision sensor, it is necessary to know how to transform data represented in one coordinate frame to another. Once the relative configuration of these two sensors are fixed, such a transformation relation between two coordinate frames of sensors can be determined via the sensor calibration process. Various algorithms can be used for this purpose such as pseudo inverse based homography estimation method, SVD based homography estimation method, and extrinsic parameter estimation method. In this research, we use a pseudo inverse based on point alignment method [38]. In the following EQUATION 1, T_I^R is a transformation matrix that transforms the radar coordinates into the image coordinates of a camera, $(x_r,\ y_r)$ is the x, y coordinates of a point from a radar sensor, and (u, v) is the coordinate of the point projected onto the image plane. In the point alignment method, the transformation matrix T_I^R can be determined by the linear least square (LS) method using a dataset obtained from both sensors for sensor calibration. To calculate the homography matrix, four pairs of stereo vision and radar sensors detection coordinates for the object are required. For this purpose, the experimental environment was constructed as shown in Figure 9, and the detection coordinates for four objects (the radar reflector with high radar reflectivity) were obtained. The final homography matrix was obtained by applying the LS method to four pairs of detection coordinates as shown in EQUATION 1.

Therefore, by multiplying the homography matrix by the coordinates of the object detected in real time through the radar sensor, the radar coordinates converted into the camera coordinate system can be obtained.

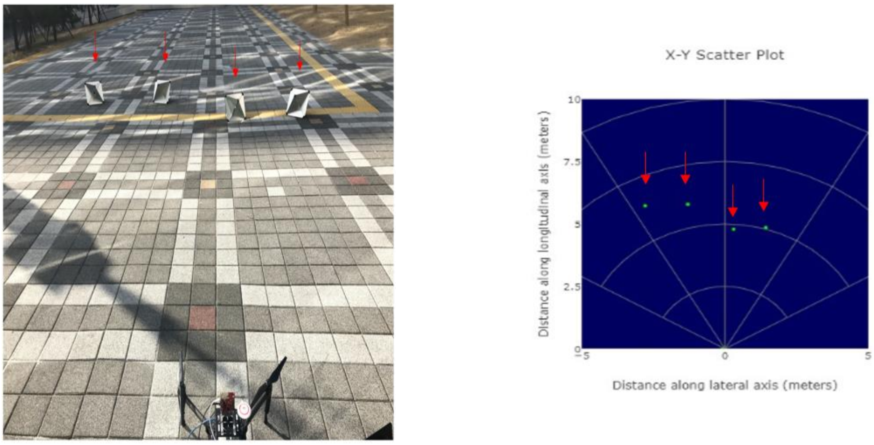


Figure 9. Experimental environment configured to obtain 4 pairs of object coordinates

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underline{T_I^R} \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} \quad \Rightarrow \quad \underline{T_I^R} = \begin{bmatrix} 356.4746 & 119.689 & 520.055 \\ 14.9625 & 15.9497 & 377.0673 \\ 0 & 0 & 1 \end{bmatrix}$$

EQUATION 1. Homography matrix calculation

4.2 UAV Detection and Tracking

One of the main contributions of the research is to implement a moving object detection and tracking system for autonomous collision avoidance of a UAV utilizing the sensor fusion between a radar and a stereo vision. For this purpose, we use YOLO v3 which is one of the deep neural network (DNN) models in Darknet, a neural network framework for learning and executing DNN, designed for real time object detection [39]. You only look once (YOLO) system is a state-of-the-art, real-time object detection system. Yolo is much faster than other object detection systems. So, making it a suitable system for real-time object detection. Conventional detection systems use a classifier to perform the detection. Applies the model to the image at different positions and scales. High score areas in the image are considered detections. However, the YOLO system uses a completely different approach. Applies a single neural network to the entire image. This network divides the image into regions and predicts the bounding box and probability of each region. This bounding box is weighted with the expected probability. The YOLO model has several advantages over classifier-based systems. At test time, the entire image is checked to inform the prediction in the global context of the image. It also predicts with a single network assessment, unlike systems like R-CNN that require thousands of single images. So, it's more than 1000 times faster than R-CNN and 100 times faster than Fast R-CNN. YOLOv3 uses several tricks to improve training and improve performance, including multi-scale prediction, better backbone classifiers, and more. Therefore, we selected the YOLO v3 system as the detection method through the stereo vision and applied the YOLO v3 system to STEREOLABS's ZED.

In addition, YOLO is an algorithm that operates in darknet frame, so darknet must first be set up on the onboard computing board. Since the mounted sensor-based UAV software system is set up based on ROS, it is essential to set up the darknet frame on the onboard computing board based on the ROS. Therefore, a ROS-based darknet frame(darknet-ROS) is set up on an onboard computing board and integrated with ZED stereo vision. And, this study aims to establish a UAV detection system by limiting the detection target to 'UAV' in order to prepare for the case where collision is expected by other UAV with malicious purpose. However, there are no 'Drone' classes in the various configurations basically provided in YOLO system. So, we use yolo-drone configuration to create a configuration that only detects UAVs. To get that weights, we built 2664 UAV image datasets and generated YOLO-drone weights through labeling and training (Figure 10). Figure 11 is the YOLO-drone system, the result of learning the mentioned above drone dataset through the YOLO system architecture.

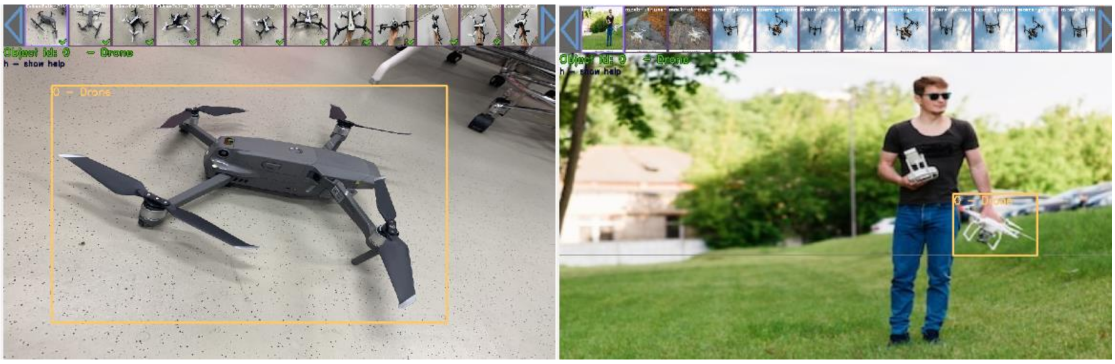


Figure 10. Drone dataset configuration & image labeling for learning

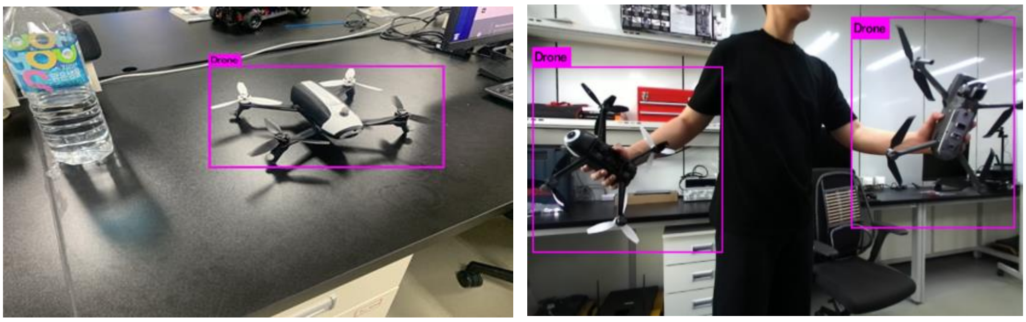


Figure 11. Detect UAV using trained YOLO-drone configuration

In addition, to compensate the mentioned above shortcomings of stereo vision-based detection system, we incorporate the measurement data from a radar sensor to improve the overall detection and tracking quality. For this, the radar detection results are projected onto the image plane of stereo vision through a coordinate transformation matrix calculated using the method described in Section 4.1 (Figure 12).

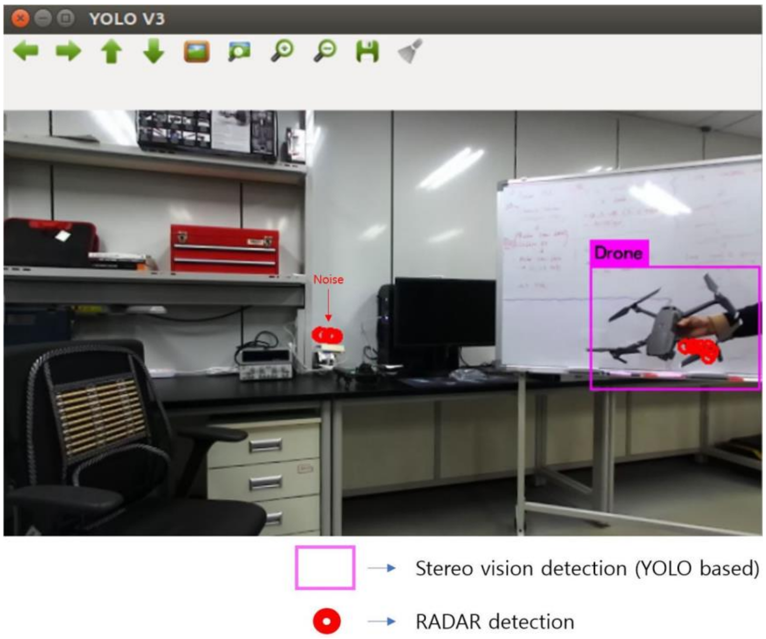


Figure 12. Project radar coordinates converted to camera coordinate system onto camera plane

Then, if the detection result of the stereo vision using the YOLO v3 algorithm matches the radar detection result for a certain object, it is classified as “strong classification”. On the other hand, if the detection result does not match or only one of the two sensors recognizes the object, the classification is classified as “weak classification”.

As described above, if the object around the UAV is recognized by the sensor fusion module, the tracking process for the detected object is essential to provide necessary information for collision avoidance algorithm. In our current implementation, objects with only “strong classification” label through sensor fusion process are selected for tracking as they have more certainty for being true objects. However, we also have a plan to study further later how to improve the accuracy of object detection in a more systematic way. For tracking, we chose the SORT [40] (Simple Online and Realtime Tracking) as a tracking algorithm. SORT algorithm a simple online and real-time tracking algorithm for 2D multiple object tracking in video sequences. SORT is the primary implementation of a visual multi-object tracking framework based on fundamental data association and state estimation techniques. Designed for online tracking applications that can only use past and current frames, this method allows to generate object IDs on the fly. SORT is an algorithm that explores practical methods for multi object tracking whose main purpose is to effectively correlate objects for real-time tracking. By changing the detector section, tracking can be improved by up to 18.9%. Based on familiar technologies such as the Kalman Filter and the Hungarian algorithm, it offers accuracy comparable to modern online trackers. Also, since the tracking method is simple, the speed increase is very high. Therefore, it is suitable as a tracking algorithm to be used for UAV detection results. paper of SORT algorithm presents a short implementation of a tracking-detection framework for the multiple object tracking (MOT) problem in which objects are detected in each frame and represented by bounding boxes. Therefore, it can be easily integrated with YOLO, a detection system that can obtain the bounding box information (coordinates of the bounding box). Unlike other tracking approaches, it targets online tracking. The focus is on effective real-time tracking and efficiency for improving the performance of applications such as pedestrian tracking. The MOT problem can be seen as a data association problem whose purpose is to correlate detection in multiple frames of video sequence. To help with data association, trackers use a variety of methods to model the objects and appearance of a sequence.

$$\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$$

EQUATION 2. Estimation model of SORT (the state of each target)

As described above EQUATION 2, the SORT algorithm describes the displacement between frames of each object using a linear constant velocity model that is independent of the motion of other objects and cameras. u is the center horizontal pixel position of the target, v is the center vertical pixel position of the target, s is the bounding box size of the target, and r is the aspect ratio of the target bounding box.

Therefore, we extracted the bounding box information of the target (UAV) to apply the SORT algorithm to our YOLO-drone system for object tracking. In order to make the velocity information more visible, we have additionally implemented an arrow indicating velocity in the bounding box of the object (relative UAV) being tracked. To verify performance of SORT algorithm, it conducted experiment by flying actual drone (Bebop 2 drone) from indoor testbed. With the M100 UAV equipped with stereo cameras and radar sensors stationary, the experiment was conducted by flying a relative drone nearby. Finally, by applying both the stereo camera-based YOLO-drone system and the SORT algorithm, the object detection and tracking system of the sensor-mounted UAV platform was established (Figure 13, 14).

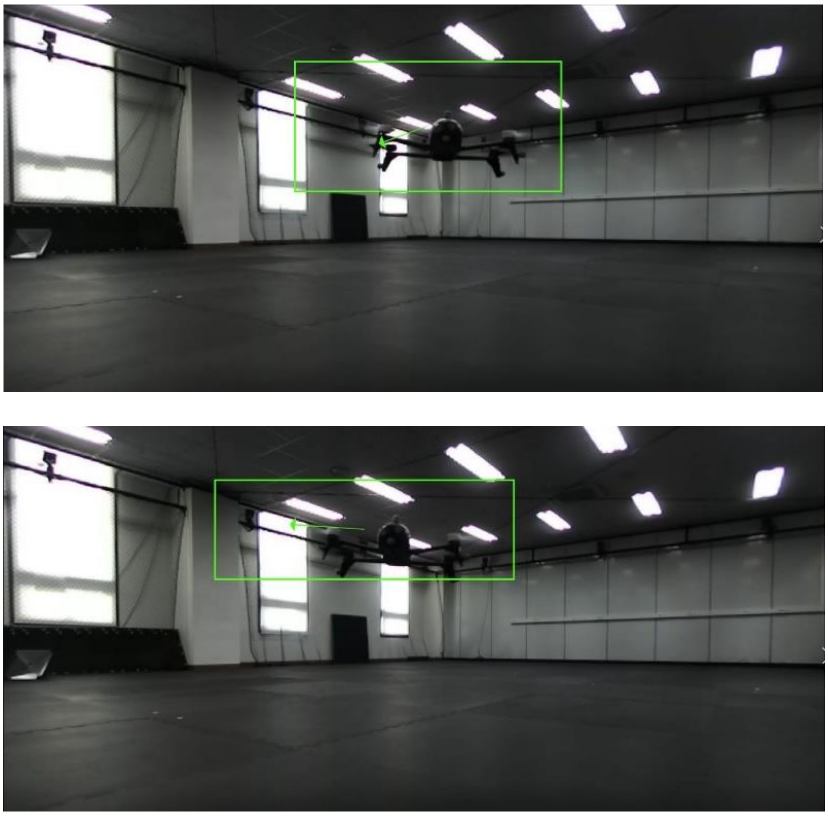
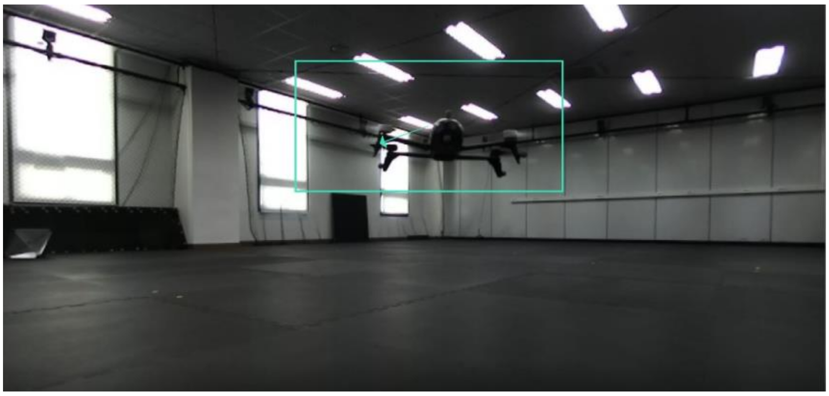
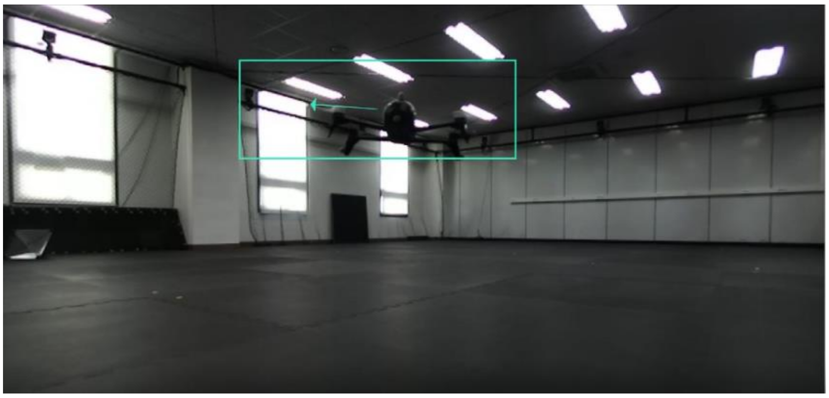


Figure 13. Experiment 1) Performance verification experiment about integration of YOLO drone detection system and SORT tracking algorithm





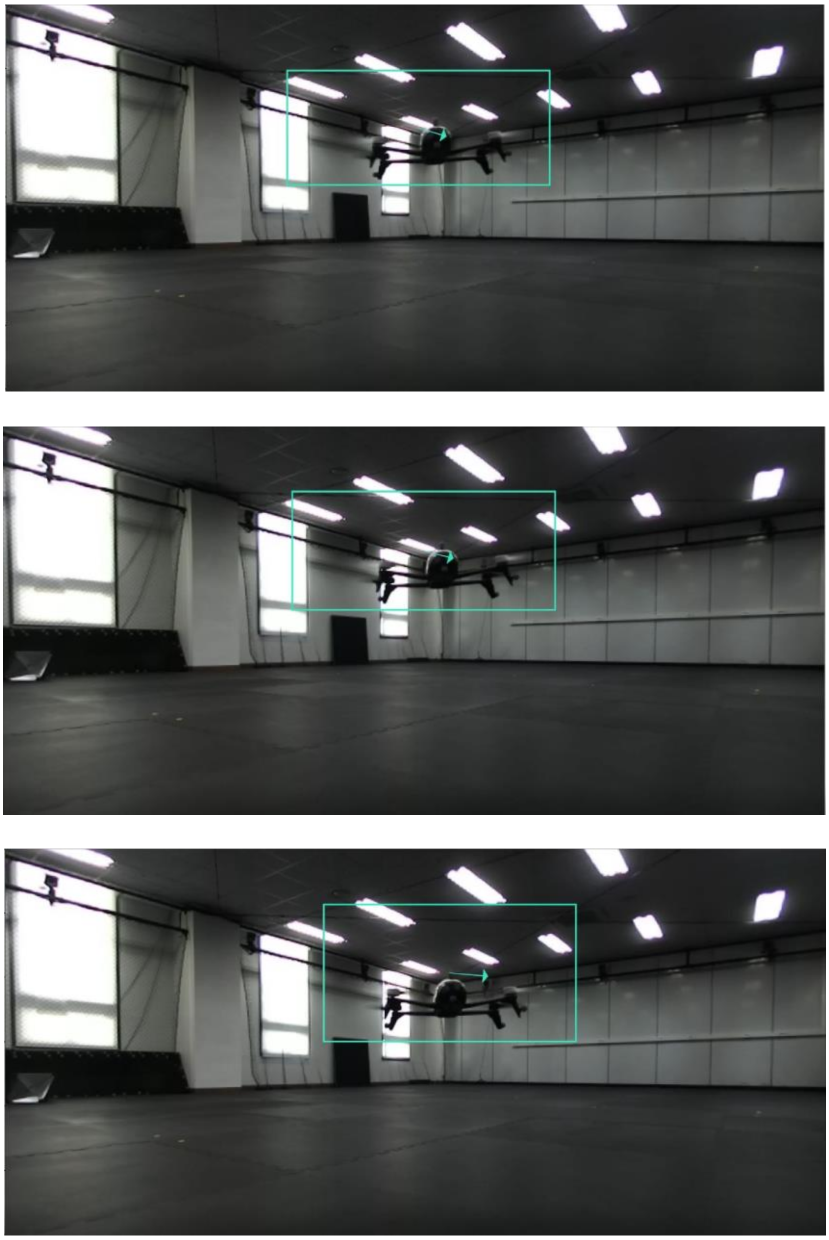


Figure 14. Experiment 2) Performance verification experiment about integration of YOLO drone detection system and SORT tracking algorithm

In the SORT algorithm, the Kalman filter used for object tracking performs repeated status prediction and measurement updates to calculate the robot's current position. The Status Prediction step is to predict current robot state values, such as location, speed, etc. Measurement updates are the steps to update the current robot status values using the current robot status values and sensors in the status prediction phase. Therefore, we implemented ‘tracking_node’1 based on SORT algorithm to track objects as above.

4.3 Local Map for Static Object Detection

Fusion of stereo vision and radar sensor makes it easier to detect moving objects. This is because radar sensors can easily detect moving objects through the Doppler effect. Radar sensors, however, do not detect static objects very well. Therefore, countermeasures against static objects such buildings, trees, etc. are necessary for the purpose of collision avoidance. To do this, we apply local mapping to get information about static objects around the UAV. The localization problem is a problem of predicting where a robot is located on a map by using a sensor mounted on the robot when a given environment map is given. For this, we construct a stereo vision based OctoMap[41] through the OctoMap mapping. OctoMap is efficient probabilistic 3D mapping framework based on octrees. Three-dimensional models provide volumetric space that is important for a variety of robotic applications, including flight robots and robots equipped with manipulators. OctoMap provides an open source framework for creating volumetric 3D environment models. The octomap mapping approach is based on octrees and uses stochastic occupancy estimates. and, It clearly refers to the occupied space as well as free and unknown areas. OctoMap also proposes an octree map compression method for miniaturizing 3D models. The OctoMap framework is available as an open source C ++ library and has already been successfully applied to many robotic projects. We determined that OctoMap mapping is suitable in environments where accurate recognition of surrounding static objects is required during flight, such as in the UAV system. Judgment reasons are due to the following advantages of OctoMap. First, OctoMap can model arbitrary environments without prior assumptions. The representation model can occupy areas as well as available space. Unknown areas of the environment are implicitly encrypted on the map. The distinction between free space and occupied space is essential for safe robot navigation, but in UAV systems, which are often used for purposes such as environmental self-exploration, information on unknown areas is more important, and OctoMap can meet that purpose. Second, new information or sensor-readings can be added at any time. Modeling and updating are conducted in a probabilistic manner. This describes sensor noise or measurement caused by dynamic changes in the environment (e.g., dynamic objects). Also, several robots can contribute to the same map, and previously recorded maps can be expanded when exploring new areas. To build an autonomous flying system for UAVs, the storage of information about previously flown maps and the ongoing calculation of new maps are essential, and the use of the OctoMap can solve the above problems. third, don't need to know the extent of the map in advance. Instead, maps are dynamically expanded as needed. and, the map is multi-resolution. that allows efficient visualization that extends from a rough overview to a detailed close-up view. Finally, the map is efficiently stored on both memory and disk. Compressed files can be generated for later use or convenient exchange between UAVs even under bandwidth constraints. PointCloud2 Topic was created using ROS_ZED_WRAPPER to implement stereo camera based OctoMap mapping mounted on a M100 UAV system. The ROS_ZED_WRAPPER package allows to use ZOS stereo cameras with ROS. Output camera left and right image, depth map, point cloud, pose information and support the use of multiple ZED cameras. and, OctoMap generate Obstacle Voxels by using PointCloud2. In order to verify the performance of the generated OctoMap, we experimented based on the TX2 board, an onboard system installed in the M100 system. The experiment was conducted largely through two scenarios, with the first being an experiment on local mapping, which verified the real-time local mapping performance using OctoMap mapping as it passed through the corridor with an M100 UAV equipped with a stereo camera and onboard computer (Figure 15). The second experiment, under the same circumstances, was conducted on global mapping using OctoMap.

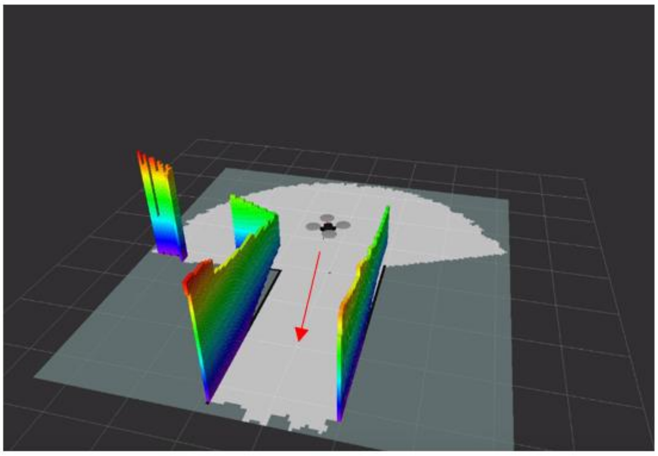


Figure 15. Experimental environment for the performance verification of the OctoMap mapping based on ZED stereo camera (move along hallway with M100 UAV)

As a local, global mapping experiment with OctoMap (Figure 16, 17), the mapping for the corridor side wall as shown in the figure was conducted, and the color represented by one voxel means depth.

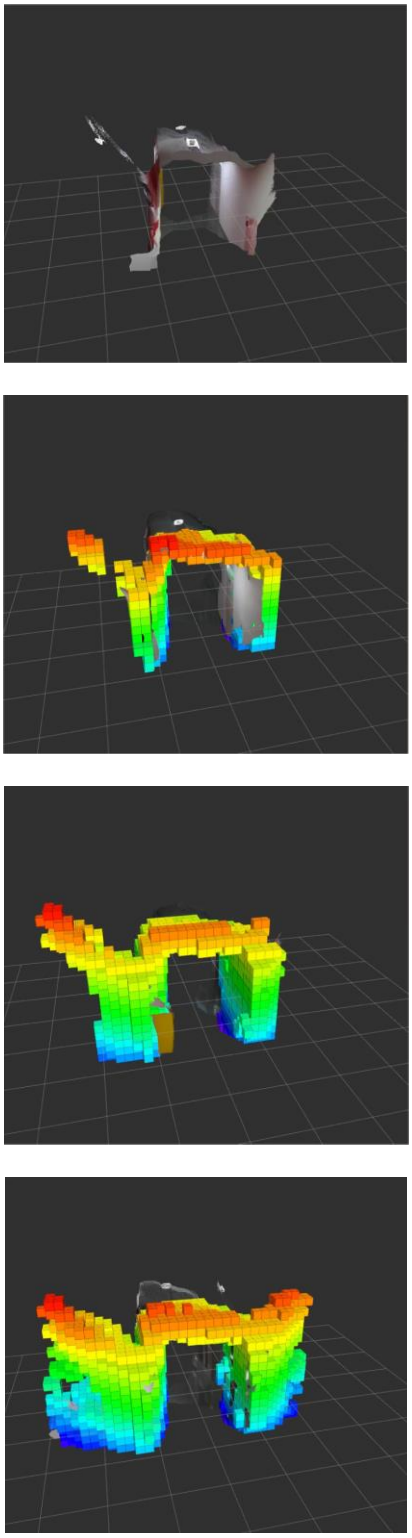


Figure 16. Local mapping process using OctoMap

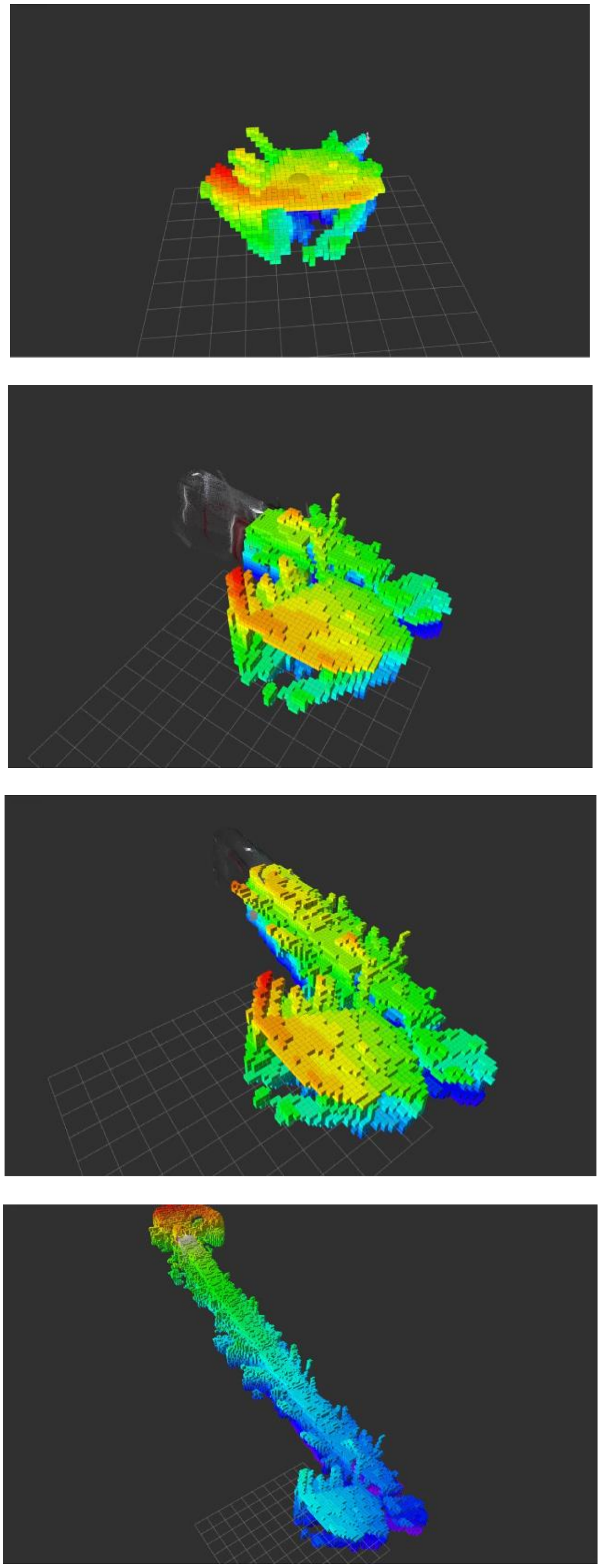


Figure 17. Global mapping process using OctoMap

V. Collision Avoidance

This section describes the development of an autonomous collision avoidance algorithm for improving the flight safety of remotely controlled UAV. for collision avoidance in teleoperation of a UAV, we will use the Vehicle-centered Potential Function (VPF) as the collision avoidance algorithm. The basic idea of the VPF, an algorithm based on potential functions, is to generate collision avoidance motion for a UAV considering the relative distance and maximum speed, and also the maximum acceleration/deceleration of objects within with environment. More specifically, VPF solves the problem of traditional RPF-based UAV autonomous collision avoidance technology. There are several problems with traditional RPF-based UAV autonomous collision avoidance techniques. First, the existing dynamic RPF generates near-infinity collision avoidance force as the surrounding object gets closer to the UAV, and such infinite collision avoidance force is not physically feasible. In this paper[42], it is proposed to go through converter block processing to avoid this problem. However, if the maximum collision avoidance force of dynamic RPF is designed to be finite rather than infinity, unnecessary process such as converter block can be omitted. Second, the method being proposed to enable the drone to follow the remote controller's input and to simultaneously avoid collision with surrounding objects is simply to add the Collision avoidance input value to the remote controller's input. However, such a simple process may not be possible to avoid collision as intended. For example, if the remote controller’s input value and the Collision avoidance input value are the same size and the direction is reversed, the drone will remain stationary, and a collision cannot be avoided if the object is moving in the direction of the UAV. Third, the collision avoidance force calculation through the RPF is performed on all objects detected in the FOV of the sensor mounted on the UAV. If the FOV of the sensor is narrow, collision avoidance is very limited, and there is a problem even if the FOV is equipped with a very wide sensor. In particular, even if the object is detected in the sensing area, it is not necessary to perform the collision avoidance response for the object that does not actually collide with the UAV. Thus, if the collision avoidance calculation process is performed on all objects detected in the sensing area without prior judgment of the potential collision possibility, it can increase unnecessary computational load on UAV that cannot be equipped with high-performance processors, resulting in reduced flight and collision avoidance performance. Finally, even though RPF is a dynamic RPF that includes both relative distance and velocity with objects, no results have been reported for avoiding flow obstacles in remote control UAVs, and only collision avoidance results for a single fixed obstacle are presented. In particular, the contents of the paper cannot be used in such an environment because there is no suggestion on how to deal with collision avoidance when there are many moving obstacles. Therefore, as a way to solve the problems mentioned above, RPF-based UAV autonomous collision avoidance technology called VPF is presented. The formula for VPF is:

$$U(\mathbf{p}, \mathbf{v}) := \begin{cases} f(\mathbf{p}, \mathbf{v}) & \text{if } d_{O/U} > d_{stop}(v_{O/U}) \\ 1 & \text{if } d_{O/U} \leq d_{stop}(v_{O/U}) \end{cases}$$

where

$$f(\mathbf{p}, \mathbf{v}) = \left(\frac{d_{O/U} - d_{stop}(v_{O/U})}{d_S - d_{stop}(v_{O/U})} - 1 \right)^2$$

EQUATION 3. Basic formula for VPF

where $d_{O/U}$ represents the relative distance between the UAV and the object, and $d_{stop}(v_{O/U})$ represents the minimum stopping distance calculated based on the relative speed between the UAV and the object. The shape of the RPF defined in this way changes with the relative velocity values between the UAV and the counterpart. Also, to avoid collisions with other objects Collision avoidance input calculations can be calculated using the gradient of the equation above (EQUATION 3).

And as you can see in Figures 18 and 19, the VPF is not dependent on the FOV of a particular sensor and can be applied to any sensor that can detect a 360-degree area around the UAV. As noted in the third problem of the preceding technology mentioned above, it is necessary to reduce the computations required for collision avoidance as much as possible, considering the difficult unmanned aircraft environment to mount high-performance computing devices. To do this, it is necessary to identify the potential for collision between surrounding objects within the detection range of the UAVs.

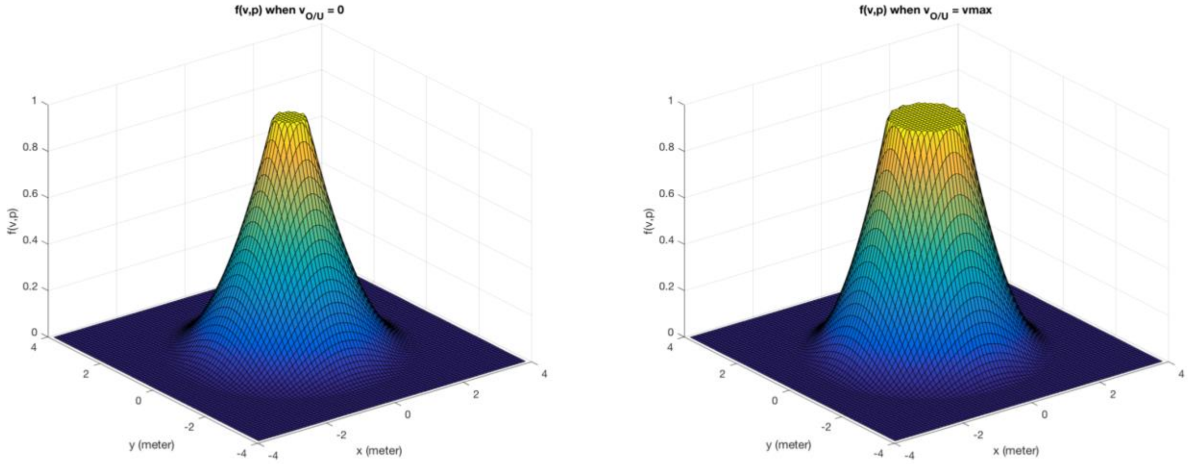


Figure 18. Dynamic Repulsive Potential Function

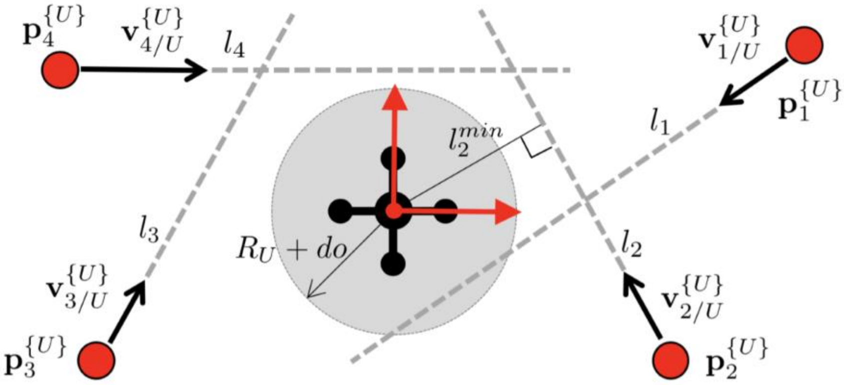


Figure 19. Potential collision discrimination technique

To this end, we propose a potential collision discrimination technique as shown in Figure 19 based on the relative distance and relative velocity between the UAV and the surrounding objects. Figure 19 shows the relative velocity and relative distance values in the fixed coordinate system of the UAV. In this situation, it is determined that there is a possibility of collision when the estimated moving path among the objects overlaps with the UAV safe zone (gray area around the UAV). Next, for autonomous collision avoidance of the remote-controlled UAV, how to fusion the input value of the remote controller and the collision avoidance input through the RPF is very important. As pointed out in the second problem of the prior technology, collision avoidance is impossible through simple sum of input values, and accurate analysis/discrimination and supplementation are necessary. In this paper, we propose the following technique to overcome this problem. First, VPF-based collision avoidance inputs for collision avoidance with neighboring objects are calculated using both collision avoidance inputs calculated based on the UAV's adjustment direction and collision avoidance inputs considering the actual flight direction of the UAV. In addition, the flight direction input value of the UAV, which is finally input to the flight controller of the UAV, is provided to the UAV by combining the collision avoidance input value and the remote controller input value previously calculated as shown EQUATION 4.

In more detail, for a given motion command from an operator (V_{OP}), the VPF-based collision avoidance algorithm generates the final motion command (V_{CMD}) for the UAV so that the UAV follows the operator's command while avoiding collisions with surrounding objects. As shown in EQUATION 4. V_{CMD} is composed of the operator's operation command and repulsive motion generated based on VPF, i.e.,

$$V_{CMD} \coloneqq V_{OP} + V_{CA}^{O}$$

where

$$V_{CA}^{O} \coloneqq \sum_{i=1}^{N} (V_{OP/CA}^{i} + V_{CA}^{i})$$

EQUATION 4. Basic formula for Sequential Collision Avoidance Motion Calculation (SCAMC)

O := $\{O^i\}_{i=1}^{N}$ is the set of objects within the UAV's sensing range R_s, N is the number of objects in O, $V_{OP/CA}^{i}$ and V_{CA}^{i} are the repulsive motion vectors for object O^i caused by V_{OP} and v respectively. When we combine ($V_{OP/CA}^{i} + V_{CA}^{i}$) for all objects in O to compute the V_{CA}^{O} in EQUATION 4, we also apply a special computation process, called the sequential computation algorithm, to generate an evasive collision avoidance motion when needed. Once the final motion command V_{CMD} is computed by EQUATION 4, then it is sent to the flight controller of the UAV making it possible a safe teleoperated navigation. In addition, as shown in the above EQUATION 4, the collision avoidance input value V_{CA}^{O} can be applied not only to single object but also to a multi objects. It is necessary to use the SCAMC technique described in more detail below. Next, in a dynamic environment (i.e., an environment in which objects around the drone move freely), a collision can occur by the movement of the surrounding object, even if the drone is stationary during flight. Collision avoidance in these cases is not considered at all in the prior art and is highly likely to cause a collision in the operation of the actual UAV system. To address these situations, the Sequential Collision Avoidance Motion Calculation (SCAMC) technique allows unmanned aircraft to automatically generate evasive actions to predict the possibility of collisions with surrounding objects and prevent expected collisions without the need for remote control operation commands. The basic process of the proposed SCAMC technique is as follows. First, looking at the above-described calculation process of V_{CA}^{O}, after the collision avoidance input values for the N peripheral objects are calculated, the final V_{CA}^{O} collision avoidance value is calculated through the vector sum. In this case, by performing the vector sum in a sequential manner as shown in the following equation, it is possible to determine whether there is a potential collision, and if a collision is expected, it is performed by adding an operation required for avoidance.

$$\mathbf{v}_{CA}^{1:i} := \begin{cases} \mathbf{v}_{CA}^{1} & \text{for } i = 1 \\ \mathbf{v}_{CA}^{1:i-1} + \mathbf{v}_{CA}^{i} & \text{for } i = 2, 3, \ldots, N. \end{cases}$$

EQUATION 5. Sequential computation of V_{CA}^{O}

In the above EQUATION 5, if the directions of the $V_{CA}^{1:i-1}$ and V_{CA}^{i} vectors are reversely closer to each other, it is possible that a probability of potential collision is high. At this time, the additional avoidance operation is generated in a direction orthogonal to the plane generated by the combination of the two vectors for collision avoidance. The following EQUATION 6 is an expression that expresses the process of creating an additional avoidance moving on the sum of two vectors for avoidance moving if there is a potential collision. k is a constant value indicating the magnitude of the avoidance action, and $(e_{CA}^{i})_{\perp}$ is a unit vector indicating the direction of the avoidance action.

$$\mathbf{v}_{CA}^{1:i} = \mathbf{v}_{CA}^{1:i-1} + \mathbf{v}_{CA}^{i} + \kappa \left(\mathbf{e}_{CA}^{i}\right)_{\perp}$$

EQUATION 6. Create additional collision evasive motion

Figure 20 shows an example of a situation where the UAV in hovering avoids a collision through the SCAMC calculation process.

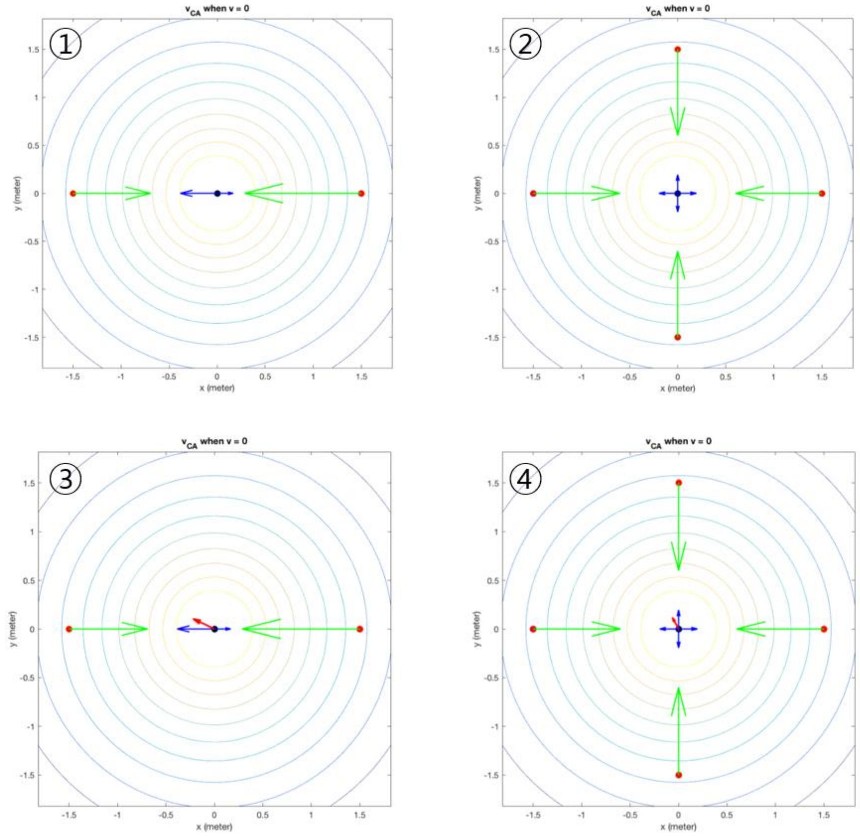


Figure 20. Generation of Collision Avoidance Motions for Hovering state UAV Using SCAMC Computation

To verify the performance of the VPF, simulation software called the Virtual Robot Experiment Platform (V-REP) was used. As shown in Figure 21 -24, we experimented with collision avoidance for dynamic objects by forming a cylinder moving along a certain path in a simulation environment and flying a virtual UAV equipped with a VPF algorithm.

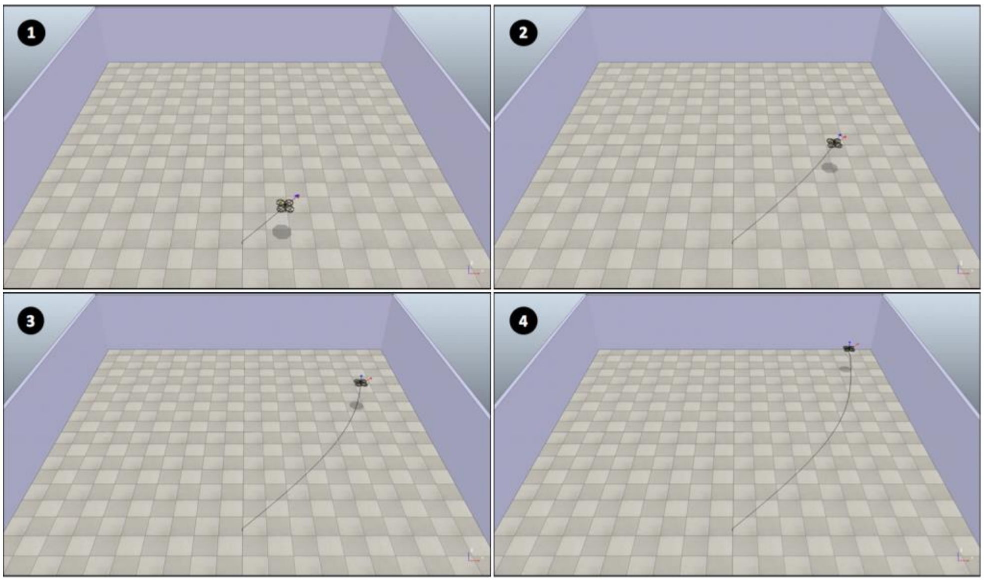


Figure 21. Collision avoidance against wall

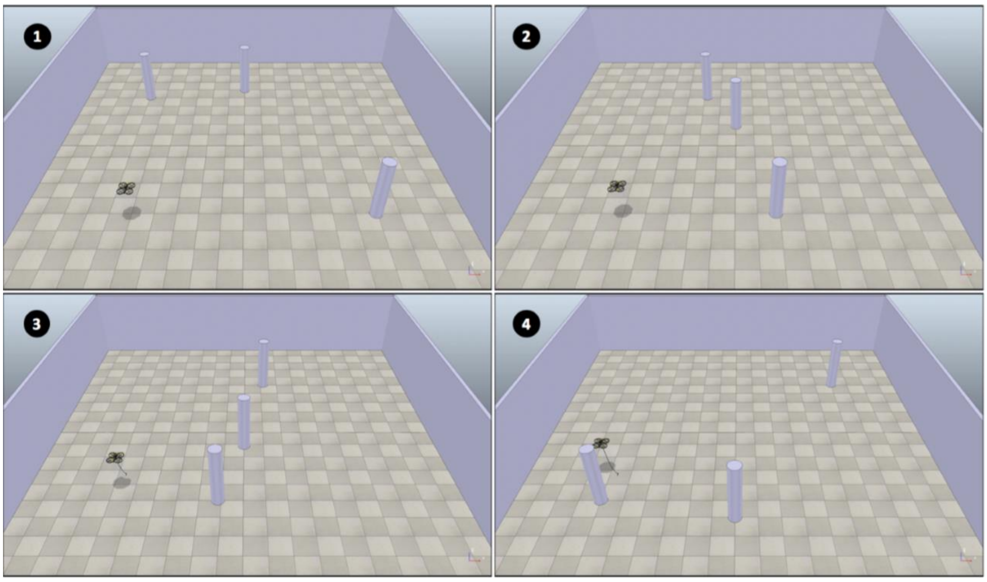


Figure 22. Collision avoidance through evasive motion created in sequential calculation of V_{CA}^{O}

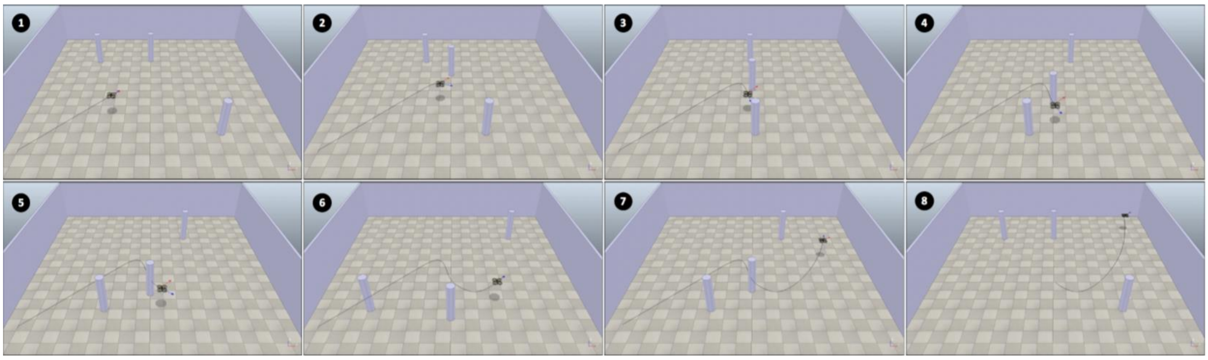


Figure 23. Collision avoidance against three moving objects when the operator's command is toward the forward-right direction

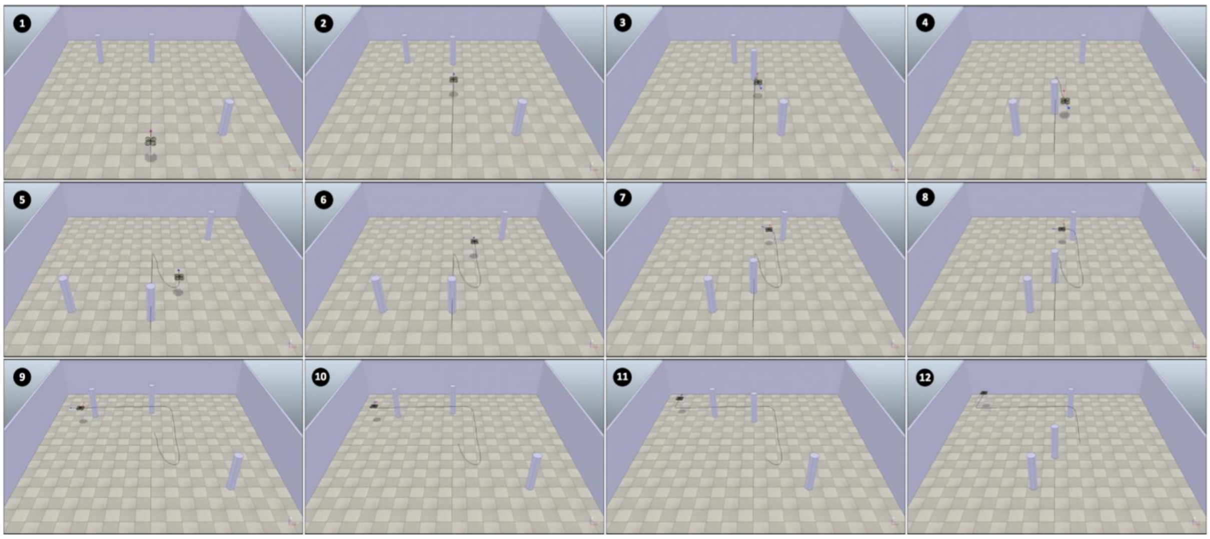


Figure 24. Collision avoidance against three moving objects when the operator's command is in forward direction directly toward one of the moving objects

VI. Conclusions

In this study, we performed three major tasks. First, the hardware and software setup of the UAV platform (M100) laid the foundation for building an autonomous collision avoidance system. Second, we built a sensor fusion module and mounted to UAV to build a perception system and verified the possibility of object detection and tracking through sensor fusion data. Third, we presented the VPF, a collision avoidance algorithm that can overcome the limitations of the existing collision avoidance algorithms and verified the performance through simulations.

The main focus of this research is to develop a UAV system for autonomous collision avoidance and perform experimentation. In this regard, we conduct experimentation to validate the performance of autonomous collision avoidance using a real UAV platform. First, sensor fusion classified into 'strong classification' and 'weak classification' to obtain more reliable detection results. After that, the SORT algorithm tracks the detected objects and finally integrates them with the collision avoidance algorithm. Finally, we used a stereo camera based OctoMap mapping to perform local and global mapping. It can get information about static objects and work with collision avoidance algorithms. but further work still needs to be done. The formed 3D local map needs to find information about the static object via VoxelGrid, and the moving object detected on the local map must be associated with a sensor fusion based detection & tracking algorithm. This allows moving objects to be detected on the local map.

References

[1] J. Tisdale, Z. Kim, and J. K. Hedrick, “Autonomous uav path planning and estimation,” IEEE Robotics & Automation Magazine, vol. 16, no. 2, 2009.

[2] R. He, S. Prentice, and N. Roy, “Planning in information space for a quadrotor helicopter in a gps-denied environment,” in Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, 2008, pp. 1814–1820.

[3] Y. Bouktir, M. Haddad, and T. Chettibi, “Trajectory planning for a quadrotor helicopter,” in Control and Automation, 2008 16th Mediterranean Conference on. Ieee, 2008, pp. 1258–1263.

[4] G. Fasano, D. Accardo, A. Moccia, C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo, “Multi-sensor-based fully autonomous non-cooperative collision avoidance system for unmanned air vehicles,” Journal of aerospace computing, information, and communication, vol. 5, no. 10, pp. 338–360, 2008.

[5] X. Wang, V. Yadav, and S. Balakrishnan, “Cooperative uav formation flying with obstacle/collision avoidance,” 2007.

[6] C. Masone, P. R. Giordano, H. H. B¨ulthoff, and A. Franchi, “Semiautonomous trajectory generation for mobile robots with integral haptic shared control,” in Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014, pp. 6468–6475.

[7] S. Stramigioli, R. Mahony, and P. Corke, “A novel approach to haptic tele-operation of aerial robot vehicles,” in Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010, pp. 5302–5308.

[8] A. Y. Mersha, S. Stramigioli, and R. Carloni, “Switching-based mapping and control for haptic teleoperation of aerial robots,” in Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012, pp. 2629–2634.

[9] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. Van Paassen, “Artificial force field for haptic feedback in uav teleoperation,” IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 39, no. 6, pp. 1316–1330, 2009.

[10] F. Rehmatullah and J. Kelly, “Vision-based collision avoidance for personal aerial vehicles using dynamic potential fields,” in Computer and Robot Vision (CRV), 2015 12th Conference on. IEEE, 2015, pp. 297–304.

[11] P. Stegagno, M. Basile, H. H. B¨ulthoff, and A. Franchi, “A semiautonomous uav platform for indoor remote operation with visual and haptic feedback,” in Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014, pp. 3862–3869.

[12] Carbone, C., et al. "A novel 3D geometric algorithm for aircraft autonomous collision avoidance." *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006.

[13] Temizer, Selim, et al. "Collision avoidance for unmanned aircraft using Markov decision processes." *AIAA guidance, navigation, and control conference*. 2010.

[14] Chisholm, Ryan A., et al. "UAV LiDAR for below-canopy forest surveys." *Journal of Unmanned Vehicle Systems* 1.01 (2013): 61-68.

[15] Lim, Kevin, et al. "LiDAR remote sensing of forest structure." *Progress in physical geography* 27.1 (2003): 88-106.

[16] Sprunk, Christoph, et al. "Lidar-based teach-and-repeat of mobile robot trajectories." *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013.

[17] Richard, Pierre-Luc, Nicolas Pouliot, and Serge Montambault. "Introduction of a LIDAR-based obstacle detection system on the LineScout power line robot." *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2014.

[18] Wallace, Luke, Arko Lucieer, and Christopher S. Watson. "Evaluating tree detection and segmentation routines on very high resolution UAV LiDAR data." *IEEE Transactions on Geoscience and Remote Sensing* 52.12 (2014): 7619-7628.

[19] Celik, Koray, Soon-Jo Chung, and Arun Somani. "Mvcslam: mono-vision corner slam for autonomous micro-helicopters in gps denied environments." *AIAA Guidance, Navigation and Control Conference and Exhibit*. 2008.

[20] Celik, Koray, Soon-Jo Chung, and Arun Somani. "Mono-vision corner SLAM for indoor navigation." *2008 IEEE International Conference on Electro/Information Technology*. IEEE, 2008.

[21] Choi, W., C. Ryu, and H. Kim. "Navigation of a mobile robot using mono-vision and mono-audition." *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*. Vol. 4. IEEE, 1999.

[22] Seemann, Edgar, Kai Nickel, and Rainer Stiefelhagen. "Head pose estimation using stereo vision for human-robot interaction." *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*. IEEE, 2004.

[23] Hager, Gregory D., Wen-Chung Chang, and A. Stephen Morse. "Robot hand-eye coordination based on stereo vision." *IEEE Control Systems Magazine* 15.1 (1995): 30-39.

[24] Satake, Junji, and Jun Miura. "Robust stereo-based person detection and tracking for a person following robot." *ICRA Workshop on People Detection and Tracking*. 2009.

[25] Pire, Taihú, et al. "Stereo parallel tracking and mapping for robot localization." *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[26] Cory, Phil, Hobart R. Everett, and Tracy A. Heath-Pastore. "Radar-based intruder detection for a robotic security system." *Mobile Robots XIII and Intelligent Transportation Systems*. Vol. 3525. International Society for Optics and Photonics, 1999.

[27] Falconer, David G., Robert W. Ficklin, and Kurt G. Konolige. "Robot-mounted through-wall radar for detecting, locating, and identifying building occupants." *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 2. IEEE, 2000.

[28] Smith, Gregory C., and Roger A. Smith. "A non-contact method for detecting on-line industrial robot position errors using a microwave Doppler radar motion detector." *The International Journal of Advanced Manufacturing Technology* 29.5 (2006): 605-615.

[29] Kam, Moshe, Xiaoxun Zhu, and Paul Kalata. "Sensor fusion for mobile robot navigation." *Proceedings of the IEEE* 85.1 (1997): 108-119.

[30] Stroupe, Ashley W., Martin C. Martin, and Tucker Balch. "Distributed sensor fusion for object position

estimation by multi-robot systems." *Proceedings 2001 ICRA. IEEE international conference on robotics and automation (Cat. No. 01CH37164)*. Vol. 2. IEEE, 2001.

[31] Choi, Byoung-Suk, et al. "A hierarchical algorithm for indoor mobile robot localization using RFID sensor fusion." *IEEE Transactions on Industrial Electronics* 58.6 (2011): 2226-2235.

[32] Sasiadek, J. Z., and Qi Wang. "Sensor fusion based on fuzzy Kalman filtering for autonomous robot vehicle." *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*. Vol. 4. IEEE, 1999.

[33] Odelga, Marcin, Paolo Stegagno, and Heinrich H. Bülthoff. "Obstacle detection, tracking and avoidance for a teleoperated UAV." *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016.

[34] Raphael Labayrade, Cyril Royere, Dominique Gruyer, Didier Aubert, "Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner," Autonomous Robots, Journal, Vol. 19, Issue. 2, 99. 117-140, 2005

[35] Gao, Hongbo, et al. "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment." *IEEE Transactions on Industrial Informatics* 14.9 (2018): 4224-4231.

[36] Long, Ningbo, et al. "Fusion of millimeter wave radar and RGB-depth sensors for assisted navigation of the visually impaired." *Millimetre Wave and Terahertz Sensors and Technology XI*. Vol. 10800. International Society for Optics and Photonics, 2018.

[37] Kim, Jihun, Dong Seog Han, and Benaoumeur Senouci. "Radar and vision sensor fusion for object detection in autonomous vehicle surroundings." *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018.

[38] G. Alessandretti A. Broggi P. Cerri "Vehicle and guard rail detection using radar and vision data fusion" IEEE Trans. Intell. Transp. Syst. vol. 8 no. 1 pp. 95-105 Mar. 2007.

[39] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[40] Bewley, Alex, et al. "Simple online and realtime tracking." *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016.

[41] Hornung, Armin, et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees." *Autonomous robots* 34.3 (2013): 189-206.

[42] Rehmatullah, Faizan, and Jonathan Kelly. "Vision-based collision avoidance for personal aerial vehicles using dynamic potential fields." *2015 12th Conference on Computer and Robot Vision*. IEEE, 2015.

요약문

원격조종 기반 쿼드로터 타입 무인항공기의 자율충돌회피 시스템 구축

일반적으로 무인항공기는 사람이 탑승하지 않아도 원격 또는 자동으로 운항할 수 있다는 장점 때문에 최근 몇 년 동안 탐색, 구조, 군용, 촬영 등 다양한 목적으로 여러 장소에서 성공적으로 사용되어 왔다. 그러나 현재까지의 기술, 인프라수준으로는 인간의 조종 없이 무인항공기 스스로 완벽하게 자율주행임무를 수행하기란 힘든 일이다. 인간의 조종능력 또한 무인항공기의 안전운행에 큰 영향을 미쳤기 때문에 조종자가 뛰어난 조종실력까지 겸비해야 했다. 게다가 어수선한 환경에서 무인항공기를 안전하게 원격 제어 하는 것은 절대 사소한 작업이 아니다. 따라서 무인항공기 자체의 미흡한 안전성, 조종자의 조종능력 등 여러 가지 사항들을 고려해볼 때, 무인항공기의 안전한 비행 시스템 구축을 위해서 무인항공기에 자율 충돌 회피 기술을 탑재하는 것은 필수이다.

본 논문에서는 원격조종 기반 무인 항공기의 안전한 자율 비행 시스템 구축을 목표로 하며 이를 위해 다음과 같이 크게 3 가지 연구를 수행한다. 첫 번째로, 기존의 충돌회피 알고리즘이 가지는 여러 한계성들을 극복하여 신뢰도 높은 비행 시스템을 구축할 수 있는 충돌회피 알고리즘인 VPF(Vehicle-Centered Potential Function)를 제시하고 시뮬레이션을 통해 성능을 검증한다. 두 번째로, 인지를 위한 센서들을 무인항공기에 탑재하고 하드웨어 및 소프트웨어 구축을 통해 충돌회피를 위한 다양한 기능들을 가능하게 하는 쿼드로터 타입의 중형 무인항공기 플랫폼을 구축한다. 세 번째로, 무인항공기에 탑재된 센서들(스테레오 비전, 레이더 센서)을 센서융합 함으로써 센서융합모듈을 제작한다. 그 이후 센서융합모듈로부터 얻을 수 있는 센서융합데이터를 통해서 고성능의 물체 탐지 및 트래킹 시스템을 구축한다.

핵심어: 무인항공기, 자율충돌회피, 충돌회피 알고리즘, 센서융합